ATARI MAGAZINE
N°34
MAI 1992

ATARI magazine

FALCON
Le vrai, le faux

DOSSIER LANGAGES
Les délices de la programmation

NOUVEAU
PALETTES ETENDUES
29 791 couleurs sur votre STE
COMPRESSION DE DONNEES
Augmentez la capacité de vos disquettes

Actualités, SOS questions/réponses, Initiation, Musique, Essai logiciels, Programmation, Graphisme, Lynx, Jeux, Portfolio, Télématique



BELGIQUE 183 FB – SUISSE 6,80 FS
CANADA $ 5,75



Le magazine des utilisateurs de STF/STE, MEGA ST, MEGA STE, TT, Portfolio, Lynx.

Directeur de la publication
Rédacteur en chef
Serge Fenez

Rédacteur en chef adjoint
Pascal Melet

Ont participé à la rédaction de ce numéro
(entre parenthèses le pseudo de leur Bal sur le 3615 ATARI)
Mathias Agopian (Humfrey)
Léopold Braunstein (Leopold),
Hervé Dudognon, Bertrand Espern,
Pierre-Jean Goulier (Annybal)
Nicolas Jordan, Al Jollyson,
Patrick Leclercq (Kendrix),
Alain Lioret (Lioret),
Thierry de Rouet (Derouet),
Alexis Valey (Prtfolio).
Télématique
Laurent Guinard.
Fabrication: Florence Nivelet
Maquette: Sophie Folliard
Correctrice: Élisabeth Mignon

Photogravure: CG Couleur
Flashage: SCAP Informatique
Impression
Berger-Levrault, ICT, Fima.

La mise en page d'Atari Magazine n°34 a été entièrement réalisée sur une station PAO Atari TT avec le logiciel Calamus. Flashage haute résolution 2400 dpi.

En couverture: Shadow of the Beast de Psygnosis

Atari Magazine
est édité par ARTIPRESSE
79, avenue Louis Roche
92238 Gennevilliers CEDEX
SIREN 345 365 191, APE 5120.

Publicité au journal: Michel Sarfati
Téléphone: (1)40.85.31.90
Télécopie: (1)40.86.29.97.

Dépot légal à parution.
Tirage : 56 000 exemplaires.
Distribution NMPP.
ISSN: 0992-2016

SOMMAIRE

ACTUALITES

2 • **Univers Atari**
Le point sur le Falcon. Les nouveautés du Music Messe de Francfort et les autres.

6 • **Dernière minute**

8 **SOS QUESTIONS/REPONSES**
Des réponses, des trucs, des astuces.

MUSIQUE

14 • **My Band**
Tout un quartet à domicile. A vous les soirées qui swinguent.

18 • **Musique assistée par ordinateur (III)**
Des messages qui feront «chanter» votre STE.

41 • **Tableaux de programmation MAO**

ESSAI LOGICIEL

22 • **Six nouveaux éducatifs**
Les vacances scolaires approchent. L'occasion de réviser en s'amusant.

28 • **Sténo 2**
Simple et convivial, un éditeur de texte qui sera toujours là quand vous en aurez besoin.

30 • **STalker**
Un ensemble de communications par modem d'une grande richesse fonctionnelle.

88 • **PC Command**
Une interface de commande type PC sur ST.

REPORTAGE

26 • **Synergie & Communications**
L'une des toutes premières sociétés à avoir développé des applications professionnelles sur ST nous rappelle l'histoire de son évolution.

JEUX

34 • **Blocs-notes**

LYNX

38 • **Nouveautés**
Toki et SuperSkweek, deux stars du jeu vidéo.

PROGRAMMATION

42 • **Initiation au C**
Des scrollings verticaux à vous couper le souffle.

48 • **Initiation au GFA**
Comment éviter les erreurs de programmation.

54 • **Initiation à l'Omikron**
Faisons tomber les masques.

GRAPHISME

57 • **Les rasters**
La nouvelle version de Neochrome Master est enfin disponible.

60 • **Palettes étendues**
29791 couleurs. De quoi en faire pâlir plus d'un!

62 • **Quick Ray Trace (V)**
Une explosion d'ellipsoïdes et d'hyperboloïdes.

DOSSIER LANGAGES DE PROGRAMMATION

66 • **Dialoguez avec votre ordinateur**
Apprendre à «parler ordinateur» un peu comme on parle anglais ou allemand. Découvrez le monde passionnant de la programmation.

70 • **Patience et longueur de temps...**
Petite histoire de la programmation.

72 • **Les langages d'initiation**
Grâce aux langages dérivés du Basic, adonnez-vous aux joies de la création informatique.

76 • **Les langages avancés**
Choisir un langage avancé, c'est avant tout bien connaître ses principales caractéristiques.

80 • **Profession programmeur**
La programmation peut aussi devenir un métier. C'est généralement la passion qui guide ceux qui le choisissent.

PORTFOLIO

84 • **Le Portfolio et son écran**
Un beau graphique vaut souvent mieux qu'un long discours.

ESSAI MATERIEL

86 • **Trans-Drive**
Transférez quatorze fois plus vite vos fichiers entre un Portfolio et un PC de bureau.

3615 ATARI

92 • **Les nouveautés du 3615 Atari**
La détente à coup sûr.

94 • **Le thème du mois: les compresseurs de données**
Ne soyez plus limité par la capacité de vos disquettes ou de votre disque dur.

EDITO

Elle court elle court la rumeur*

Dans toute la presse micro, faux scoops et vraies affabulations se succèdent à un rythme soutenu: la nouvelle merveille Atari pointe le bout de son nez! Eh bien non! Nous sommes désolés de refroidir ce bel enthousiasme, mais il n'en est rien.

Rappelons les faits, et tenons-nous en aux faits, M. Jack Tramiel a annoncé à l'occasion du CeBIT de Hannovre, le mois dernier, qu'Atari Corp. avait consacré plusieurs millions de dollars à la conception d'une nouvelle gamme d'ordinateurs Atari. Peu de caractéristiques ont été avancées, et toujours du bout des lèvres, pour la bonne raison que des choix technologiques et économiques fondamentaux n'ont pas encore été confirmés.

Il faut du temps pour passer du concept au produit fini et le Falcon n'est pas exempt de cette réalité, mainte fois vérifiée, surtout dans le domaine de la micro-informatique.

Dès que nous aurons connaissance des caractéristiques de la gamme Falcon, nous vous en ferons part (et vous savez qu'Atari Magazine est bien placé pour vous donner les **vraies** caractéristiques); quand nous aurons testé un Falcon vous aurez droit à un article circonstancié.

En attendant, vous pouvez constater à la lecture de ce numéro que le marché de la gamme ST est toujours aussi dynamique: matériels et logiciels continuent d'affluer.

Voici un vrai scoop: pour utiliser tout le potentiel de votre ST/STE, lisez Atari Magazine chaque mois. *S.F.*

**merci à MM. Edgard Morin et Jean-Noël Kapferer.*

UNIVERS ATARI
L'actualité du mois

Les rumeurs les plus folles circulent à propos du Falcon! Voici en exclusivité un point précis sur l'état des lieux. Outre-Rhin, le Music Messe de Francfort faisait lui aussi l'actualité.



FALCON: le vrai et le faux

Lors du dernier CeBIT (voir n°33), **Atari** a présenté un «concentré» des technologies sur lesquelles ses ingénieurs planchent actuellement. **Atari** a en effet investi plusieurs dizaines de millions de dollars dans une nouvelle gamme d'ordinateurs baptisée Falcon. La machine, montrée aux journalistes, portait la mention Falcon/030. Depuis, que d'encre a coulé dans les colonnes de la presse informatique française et étrangère, et coulera probablement encore quelque temps, d'autant qu'**Atari** demeure particulièrement silencieux, ce qui n'est pas, il faut bien le reconnaître, dans ses habitudes.

«Il n'y a pas de fumée sans feu» dit-on, mais ce n'est pas pour autant qu'il faut prendre toutes les rumeurs au pied de la lettre. Nous avons compilé toutes les informations publiées afin de discerner le probable de l'improbable.

Commençons d'abord par éplucher les informations officiellement données par **Atari** lors de différentes conférences de presse:

- Le Falcon/030 est le premier représentant d'une nouvelle génération d'ordinateurs Atari. Il est basé sur un 68030 cadencé à 16 MHz et dispose de capacités sonores et graphiques supérieures.
- Le terme «TRUE COLOUR» a été prononcé.
- Cet ordinateur pourrait apparaître dès la fin 1992.
- Il fonctionnera sous MultiTOS, la version multitâche du TOS en cours de développement chez **Atari**.

Maintenant, la presse internationale fournit de nombreux autres détails souvent contradictoires:

- Les résolutions graphiques seraient celles du TT, donc compatibles VGA.
- Palette de 16 millions de couleurs pour certains, 256000 pour d'autres. A notre connaissance, le GEM ne gérant que des images 16 bits, la deuxième hypothèse semble la plus probable, d'autant que le terme True Colour ne signifie pas 16 millions de couleurs, comme on le croit souvent, mais que chaque point de l'image contient les caractéristiques de sa couleur et non une référence à une palette.
- Plusieurs canaux sonores en stéréo qualité CD. Ce qui impliquerait un son sur 16 bits à 44.1 KHz.
- Présence d'un processeur DSP. Cette rumeur vient du fait que **Motorola** aux USA a annoncé qu'**Atari** s'intéressait de près à la technologie du DSP. Certains en ont rapidement conclu que le Falcon/030 disposerait en standard de ce processeur. Une conclusion somme toute un peu hâtive. Cela semble peu probable si le Falcon est une machine grand public, mais on peut toujours rêver...

Quel sera donc le profil final du Falcon/030? Bien inconscient celui qui prendrait aujourd'hui des paris sur les choix technologiques retenus par **Atari**.

D'autant que, pour l'instant, il n'existe qu'un vague prototype de laboratoire et qu'**Atari** a, ces dernières années, mis toujours beaucoup de temps pour passer du prototype à la fabrication série (plus de deux ans pour le TT, plus d'un pour le ST Book qui n'est toujours pas là).

On risque donc de voir encore pendant quelque temps beaucoup de fumée. Un conseil, ne prenez pas pour argent comptant tous les scoops et autres exclusivités que vous avez pu lire ou lirez prochainement, car la seule vérité est qu'**Atari** n'a encore rien présenté d'officiel... Lorsque ça se fera, Atari Magazine vous en informera!

Réseau

La société **Multipoint** annonce la disponibilité du logiciel ITOS qui permet de faire fonctionner la carte Ethernet sur TT 030 ainsi que sur MEGA STE. Intérêt? Disposer d'un réseau rapide pour 1000 F HT.

Sur les ondes

Le 14 avril dernier, Michel Rousseau consacrait son excellente émission dédiée au monde de l'informatique, «Digitales», à Atari Magazine et à l'univers Atari dans son ensemble. L'occasion, avec notre collaborateur Pascal Melet, de faire le point sur les qualités et les possibilités du ST.

«Digitales», tous les jours sur **Réussir FM** (107.2) de 12h05 à 12h25.

Gestion de cabinet dentaire

La société **Eurosoft** propose un nouveau logiciel de gestion de cabinet dentaire. Entièrement paramétrable et utilisable à plusieurs grâce à un système de mots de passe, *Chirdent* se compose de neuf modules imbriqués: suivi des patients, suivi de pharmacie, suivi d'encaissement, gestion de la clientèle, module d'utilitaires (numéroteur automatique, éditeur de radios numérisées, etc.), suivi des données, éditeur de bilans graphiques, serveur minitel et comptabilité.

Une version de démonstration (limité à la gestion de 500 patients) est d'ores et déjà disponible en téléchargement sur le 3615 ATARI.

Un «Coktel» pour étancher votre soif de connaissances

La société **Coktel Vision** annonce de nouveaux produits dans sa gamme «Jeux et Educatifs».

Jeux, tout d'abord, avec la sortie de *Bargon Attack*, jeu d'aventure et d'action où il vous faudra avoir «des nerfs d'acier, des réflexes de pistoleros et le sens de l'observation pour être le grain de sable qui enrayera la machine de guerre bargonienne...»

Suivront deux autres jeux, *Ween Prophecy* (début mai) et *Incas* (en Septembre).

Côté éducatif, la très sympatique gamme *ADI* poursuit son essor, avec une nouvelle série pour les lycéens appelés à passer leur bac: *Adibac*.

Pour les enfants de quatre-cinq ans et six-sept ans **Coktel Vision** étoffe sa gamme *Adibou* de nouveaux produits concernant l'apprentissage de la lecture, du calcul et de l'histoire.

Outre ces nouveautés, la collection *ADI* va bientôt s'enrichire de trois nouveaux packs qui sortiront en septembre 92: Application anglais primaire, Collèges application sciences naturelles (pour les classes de 6ème, 5ème, 4ème et 3ème), et Collèges applications allemand.

Un nouveau magasin de MAO à Grenoble

Les sociétés **Rythm'n Soft** et **MIDI Tech** ont ouvert le 2 mai dernier un magasin de 130 m² spécialisé dans l'informatique, la musique et le multimédia.

Disposant d'un show-room de démonstration de tous les logiciels, ordinateurs et synthétiseurs, ce magasin proposera tous les samedis des stages de formation sur différents thèmes liés à la MAO (musique assistée par ordinateur) et au multimédia.

Rythm'n soft Miditech, 4 rue Emile Augier, 38000 Grenoble.
Téléphone: (16)76.40.52.70.

Lancer de rayon pour Studio Effect

La société **Eurosoft** annonce la disponibilité prochaine d'un nouveau logiciel de ray-tracing, *Studio Ray Trace*, qui se présentera sous la forme d'un module indépendant de Studio Effect.

Nous attendons ce logiciel pour le tester et vous en reparler.

Scanner Canon sur TT

Il est désormais possible d'utiliser le scanner IX 30F Canon sur TT, grâce au logiciel *Studio Scan* de la société **Eurosoft.**

Ce logiciel très performant, dédié à la numérisation d'images (monochrome et couleur), gérait déjà l'ensemble de la gamme Epson (GT1000, 4000 et 6000). Il gère désormais le scanner Canon qui se branche directement en SCSI sur la sortie du TT.

Le jeu vidéo s'expose

La Foire de Paris, qui se tient du 29 avril au 10 mai au Parc des Expositions - Hall 1 - de la porte de versaille à Paris, ouvrira cette année ses portes aux jeux vidéos. Atari y sera fortement présent avec un stand **Atari/Ultima** qui proposera toute la gamme des logiciels disponible à ce jour. Avis aux passionnés...



Un nouveau Superbase

OXXI vient de mettre en vente sur le marché américain une nouvelle version de *Superbase Pro 3* en attendant *Superbase 4*.

OXXI a en effet racheté tous les droits et sources de la version Atari, et préside désormais à la destinée du plus célèbre des SGBD programmables sur ST.

La version de *Superbase Pro 3* comporte un nouvel éditeur de programme et de nombreux bugs ont été corrigés. Quant à la version 4, elle est attendue pour la rentrée. Reste la question: qui va se charger de l'importation de ces nouvelles versions, **Micro Application** n'ayant pas l'air d'être très intéréssée?

PowerPoint Software

Cette jeune société américaine lance un nouveau gestionnaire de réseau exploitant aussi bien l'interface LAN des MEGA STE et TT que les prises MIDI ou la RS232 ou encore les cartouches LanTech inconnues en France. PowerNet est un réseau économique, très optimisé afin de tirer le maximum de l'interface LAN, permettant à une bonne quinzaine de postes de se partager disques durs et autres périphériques.

Selon la rumeur, **ALM** serait sur les rangs pour l'importer.

Warp 9

Vous connaissez *Quick ST*, l'accélérateur logiciel? Les sources de ce programme ont été revendus par leur auteur, en décembre, à la société **Codehead.** Ces derniers ont, durant trois mois, travaillé très dur à l'amélioration de ce programme. Finalement, les plus sont tellement nombreux que **Codehead** a décidé de changer le nom du programmme.

Ce qui aurait du être *Quick ST 4* se nomme désormais *Warp 9* et vient de sortir aux Etats-Unis.

Warp 9 est plus rapide que *Quick ST* et *Turbo ST*, compatible avec tous les logiciels y compris FSMGDOS, *Touch UP*, *PPM*, *Calamus*, etc. Il fonctionne dans toutes les résolutions y compris celles du TT. L'accélération en mode VGA TT est d'ailleurs impressionnante. Il offre, en plus de nombreuses améliorations au GEM, dont des boîtes de dialogue volantes avec raccourcis clavier et ceci dans tous les logiciels GEM.

Warp 9 devrait être distribué en France par **Arobace.**



Music Messe de Francfort

Parallèlement au Cebit de Hanovre, le Salon international de la musique de Francfort présentait toutes les nouveautés en matière d'informatique musicale. L'Allemagne est particulièrement prolixe dans ce domaine, et cette année de nombreux produits, soft et hard, sont venus agrémenter la panoplie des ataristes musiciens traditionnels (sur ST/STE), mais aussi de ceux qui, avec un PortFolio, désireraient se lancer dans ce domaine passionnant.

L'enregistrement audionumérique

Pour les amateurs d'enregistrement numérique et de direct-to-disk, **Yamaha** a innové en présentant le CBX-D5, un processeur d'enregistrement audio-numérique ayant la particularité de pouvoir fonctionner conjointement avec le séquenceur *Cubase* de **Steinberg.** Le son audionumérique enregistré sur les quatre pistes (enregistrement 2 pistes, lec-

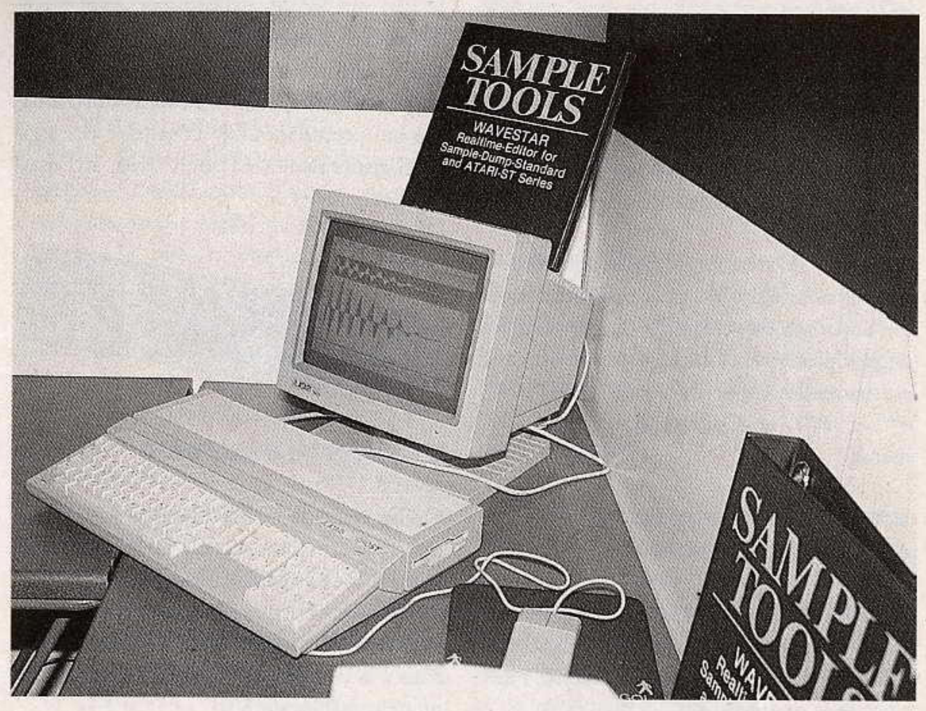


pagnée d'une batterie de logiciels (MPF.EXE), elle permettra aux utilisateurs de travailler sur des fichiers de séquence au format Midifile, d'avoir l'équivalent logiciel d'un Patchbay MIDI (permettant de mélanger des données MIDI) et de travailler sur les codes MIDI grâce à un programme du type MIDI Moniteur. Un logiciel permet le transfert des données sur ST/STE/TT. L'interface MPF 128 comporte 2 entrées et 2

(jusqu'à 520 Mo) ou d'un Syquest 44 ou 88 Mo. Le tout carrossé pour s'intégrer agréablement à un STE ou à un TT.

Echantillonnage

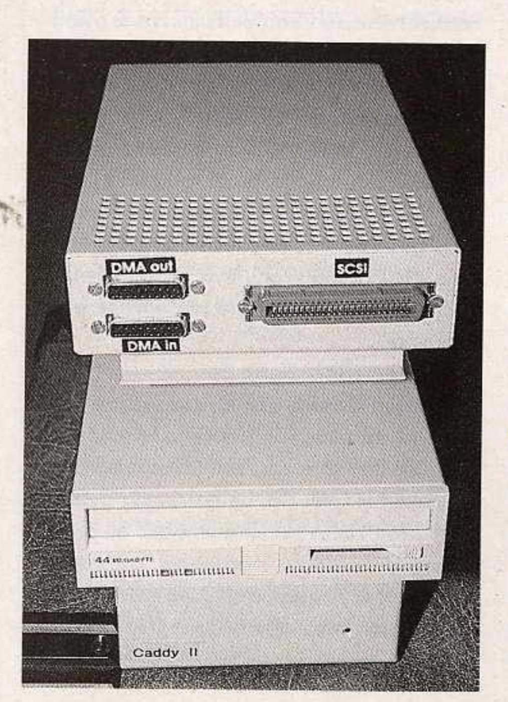
Venant d'Italie, la société **Intersound & Soft** propose toute une panoplie de logiciels permettant de travailler sur les échantillons par liaison SCSI. *Polystar* permet de charger ses échantillons à partir de vos sampleurs préférés, et de les organiser. Dix-huit mille échantillons peuvent être organisés, avec un algorithme de recherche sémantique, dans moins d'un méga de mémoire, et modifiés grâce à un éditeur temps réel. Les échantil-



ture 4 pistes) du CBX-D5 sera parfaitement synchronisé avec la séquence, les synthés et les expandeurs reliés au ST. Ce système astucieux laisse toute indépendance à l'ordinateur, puisque le boîtier CBX-D5 Yamaha effectue tout le traitement audio sans faire appel aux ressources et à la mémoire de celui-ci. Les traitements audio, les effets, et les mixages sonores sont effectués par le CBX-D5 qui sera relié par SCSI aux disques durs nécessaires. Ce système peut fonctionner sur un ST avec un minimum de 2 mégas de mémoire vive. Le CBX-D5 devrait être disponible vers la fin de l'année à un prix avoisinant les 20 000 F.

Un PortFolio qui swingue

Présenté par la société **Synthax**, voici une interface MIDI destinée au PortFolio. Accom-



sorties MIDI. Une mémoire de 128 Ko peut être ajoutée en option. Deux autres interfaces seront proposées cet été: une interface permettant toutes les synchronisations SMPTE, et un module synthétiseur qui rendra le PortFolio entièrement autonome. Il deviendra alors la première Workstation musicale autonome et facilement transportable, compatible PC et Atari.

Synthax, Stockelsdorfer Weg 1, 2407 Bad Schwartau Allemagne.

Interfaces DMA/SCSI

La société allemande de Dortmund, **MCS Midi & Computer System**, présente une gamme de disques durs équipée DMA/SCSI allant de 44 Mo à 210 Mo. Mais le plus intéressant, c'est l'interface «Caddy», composée de l'interface DMA/SCSI avec connecteurs entrée/sortie et d'un disque dur optionnel,

lons peuvent être joués par le moniteur audio du ST, ou par l'intermédiaire de la carte **Steinberg** 16-bit DA-Board.

Pour Akai S-1100, S-1000, Roland S-770, S-750, compatible Sample Dump Standard et format Avalon.

D'autres logiciels Atari permettent d'éditer et de trier pratiquement tous les formats actuels de sons échantillonnés.

Steinberg/C-Lab

Un épais public se bousculait autour de magnifiques stands (on pourrait dire des «cités» informatiques) présentés par les deux plus grosses sociétés allemandes de logiciels, **C-Lab** et **Steinberg**. Chez la première, la nouveauté c'est *Polyframe* et *Monoframe*, deux éditeurs universels qui ont provoqué l'enthousiasme des foules, ainsi que le nouveau séquenceur *Notator Logic*, qui est un

nouveau concept de séquenceur virtuel. *Notator* version 3.12 était présenté. Il inclue la possibilité de synchroniser un studio complet, équipé d'un magnéto 8 pistes **Fostex**, grâce aux nouveaux codes MIDI Machine Control.

Steinberg présentait de nombreux nouveaux logiciels, allant de *Cubase Lite* (un *Cubase* extraordinaire pour les débutants) à *Cubase Audio* intègrant l'audio numérique, en passant par des éditeurs de scores, arrangeurs et interfaces DMA/SCSI et synchro SMPTE SMP-II. La nouvelle version 3.0 de *Cubase* porte ce célèbre séquenceur au niveau des meilleurs logiciels mondiaux.

Les arrangeurs

Dans le domaine des arrangeurs, *Band In A Box* en est à la version 5.0. *Session Partner* est un nouvel arrangeur de la société **DVPI**, permettant la génération automatique de grooves et de séquences sur les quatre parties de l'arrangement (rythme, accords, basse et mélodie). Compatible Midifle, il fonctionne sur ST, STE et TT avec 1 Mo de mémoire.

Zadok présentait *Arranger ST*, un programme temps réel d'arrangement automatique sur ST, pouvant gérer 16 instruments simultanés. **Stopper Innovative Software** introduit *Presto*, un nouveau type de séquenceur de recherche incorporant une interface graphique intuitive et interactive.

La société **EMC**, qui possède un catalogue extrêmement complet d'éditeurs de synthés sur Atari, est la première à publier un éditeur pour Sound-Canvas **Roland**. Parmi les nouveautés qu'elle propose, on trouvera aussi un éditeur pour SY22/TG33, et *The Mapper*, compatible M-ROS et Soft Link, pour gérer toutes les banques de percussions. (Importé par la société **MPI**). **Soft Art**, très actif en Allemagne, mais hélas non importé en France, présentait sa collection très attrayante de logiciels pour ST et TT: didacticiels, séquenceurs, éditeurs de scores, éditeurs de sons, ainsi que le MIDI Sample Translator pour les échantillons en 12 et 16 bits.

Rectificatif

Une erreur s'est glissée dans nos actualités du mois dernier concernant le logiciel Emulcom 3.05. Atari France est bien évidemment toujours éditeur de ce logiciel.
La société Arobace est un distributeur pratiquant actuellement des prix alléchants pour ce produit.

Dernière minute!

Timeworks Publisher 2 arrive en France

La version française de Timeworks Publisher 2 pour Atari sera disponible aux alentours du 15 mai prochain. C'est la société Arobace qui en assurera la distribution en France. Son prix sera de 1 430 F.

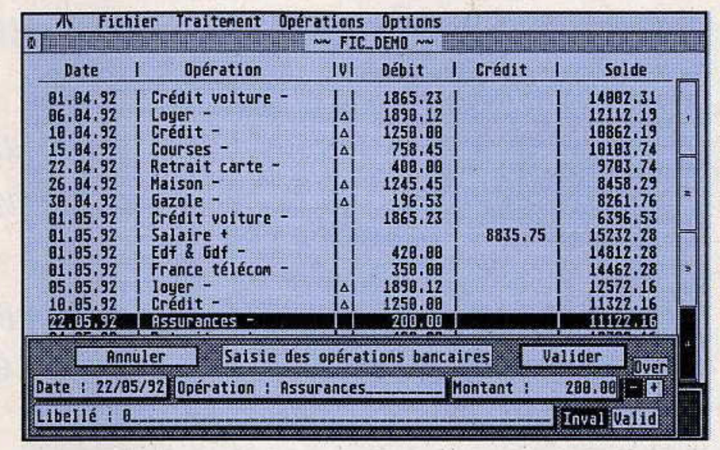
Enquête lecteur: encore un peu de patiente

Devant le nombre considérable de réponses que nous avons reçu pour l'enquête parue dans Atari Magazine n°32, nous vous demandons encore un peu de patience pour nous permettre d'en terminer le dépouillement qui se révèle plus long que nous ne l'avions escompté. Dès le travail terminé, nous procèderons au tirage au sort et à la distribution des cadeaux qui vous ont été promis.

DERNIERE MINUTE
Mieux gérer ses comptes

Fini les trous dans votre budget. Pour que vos comptes bancaires soient tenus au jour le jour, la société Logisoft propose deux nouvelles versions de STCompte et STBudget.

Les banques mettent à la disposition de leurs clients des services sur Minitel leurs permettant de connaître l'état de leurs comptes au jour le jour. Ces services sont non seulement souvent payant, mais surtout, ils ne fournissent à l'utilisateur que le solde pointé (constaté) de son compte. Or, comme chacun sait, il existe généralement une différence notable entre ce solde constaté et le solde réel d'un compte à un moment donné (différence d'autant plus grande que vous utilisez une carte bancaire à débit différé). Autre considération importante: la comparaison des recettes et des dépenses d'une période à l'autre permet de mieux comprendre la structure et l'évolution d'un budget. Par exemple, «le panier de la ménagère», sur lequel est basé l'indice des prix, omet de prendre en compte de nombreuses variables comme le prix du logement (valeur d'achat et valeur locative). Dès lors, on comprend bien l'intérêt qu'il y a à utiliser des logiciels comme *STCompte* et *ST-Budget*, qui vous permettront d'étudier, avec plus de précision, vos comptes et vos budgets.



STCompte est un programme de gestion de comptes bancaires, à l'environnement agréable, qui gère un seul compte à la fois.

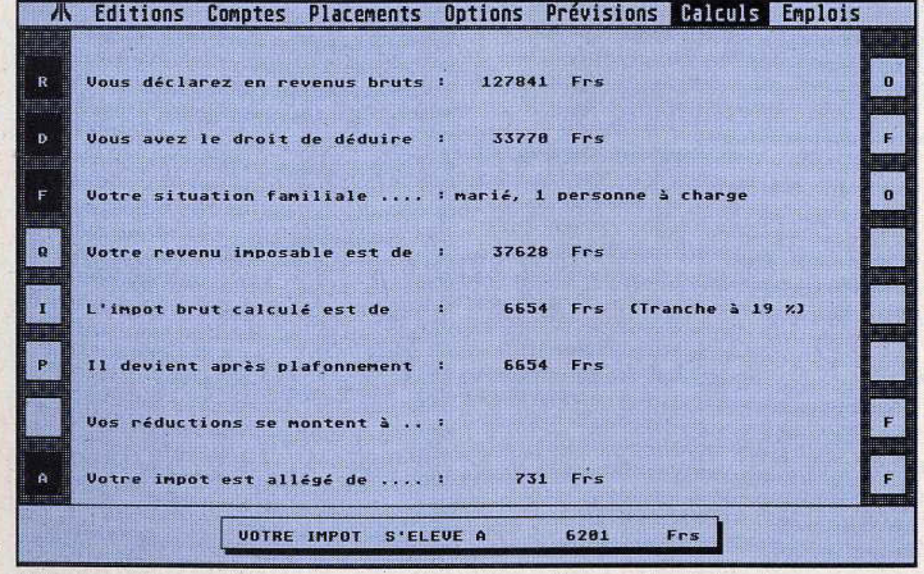
Deux produits complémentaires

STCompte, version évoluée de l'un des tout premiers gestionnaires de comptes pour Atari (*Atacompte*), passe à une version 2. *STBudget*, qui propose chaque année une kyrielle de nouvelles fonctions, inaugure le troisième volet de son développement.

Loin de se concurrencer l'un l'autre, ils offrent chacun une analyse particulière. *ST-Compte* est destiné à gérer ses relevés bancaires en prenant, pour seule base, vos dépenses et vos recettes, tandis que *STBudget* tire parti de toutes les opérations courantes d'un ménage.

Les résultats obtenus ont une base commune: le solde des comptes et l'analyse par affectation des recettes et des dépenses. Mais *STBudget* va plus loin que *STCompte* en fournissant des résultats plus pointus. A analyse particulière, orientation différente.

STCompte et *STBudget* n'ont rien de commun. Par ailleurs, ils peuvent se révéler complémentaires par le biais de l'utilisation des fonctions d'importation et d'exportation.



STBudget dispose de nombreuses fonctions qui permettent de suivre l'évolution de votre budget: ici la possibilité de calculer l'IRPP.

De la saisie à l'analyse

Depuis notre dernier comparatif (voir Atari Magazine n°26), nous pouvons constater quelques améliorations ou différences notables.

Au niveau interface utilisateur, *STCompte*

connaît de nets progrès. *STBudget*, quant à lui, est toujours aussi austère.

L'ergonomie du premier tire pleinement partie des boîtes de dialogues et des fenêtres GEM, le second ignore le GEM au profit de cadres graphiques qui ne sont pas toujours très pratiques à utiliser.

En revanche, en ce qui concerne les fonctions, *STBudget* fait meilleure impression que *STCompte*. Ainsi, le premier propose la gestion de plusieurs comptes bancaires, la gestion et le calcul de la rentabilité des petits placements (actions, SICAV ou FCP), le calcul de l'IRPP (Impôt sur le revenu des personnes physiques) ou encore des présentations d'états détaillés.

Le second autorise la gestion de plusieurs comptes, à condition de ne pas effectuer d'opérations compte à compte. Il n'est alors pas possible de gérer les dépenses effectuées en liquide ou les virements réalisés entre deux comptes courants.

C'est ici une limite importante qui n'existe pas au sein de *STBudget*. Même si ce dernier ne présente pas cet inconvénient, en revanche, la saisie des opérations reste fastidieuse. Aussi la complémentarité de ces produits s'explique par leurs limites respectives: *STCompte* est plus destiné à la saisie et au pointage des recettes et des dépenses, alors que *STBudget* est plus puissant dans ses analyses.

N'en déduisons pas à la hâte que ces deux outils ne sont pas utilisables séparément, loin s'en faut. *STCompte* permet de réaliser des statistiques chiffrées ou graphiques des recettes et des dépenses.

STBudget gère, quant à lui, les prélèvements automatiques.

Ces logiciels présentent néanmoins quelques lacunes: l'impression reste sommaire dans les deux cas, les imprimantes à jet d'encre d'entrée de gamme comme la BJ10e sont ignorées. Au final, on regrette presque qu'ils ne fusionnent pas pour devenir un seul et même outil: pourquoi pas?

Thierry de Rouet ■

STCompte 2
Edité et distribué par
Logisoft
Pour ST/STE et MEGA STE
Couleur et monochrome
Prix: 290 F

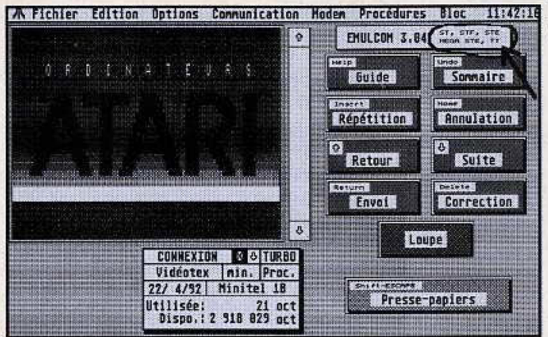
STBudget 3
Edité et distribué par
Logisoft
Pour ST/STE et MEGA STE
Couleur et monochrome
Prix: 390 F

SOS Questions/réponses

LOGICIEL

A quand une version d'Emulcom 3 fonctionnant sur TT?

Elle est d'ores et déjà disponible!
Envoyez votre disquette originale d'Emulcom 3 à Atari France (79 avenue Louis Roche, 92238 Gennevilliers) qui vous renverra la dernière version compatible TT et MEGA STE.



La version 3.04 d'Emulcom. Compatible TT et MEGA STE.

Je possède un 1040 STE. Avec *Calamus*, quand je charge une image au format .PI3, .PC3, Stad ou Calamus (.CDK), le programme me marque «Erreur système imprévisible» et, pour sortir de la boîte de dialogue, je suis obligé de cliquer sur «Fin» après quoi je retourne donc au GEM.

Vous utilisez de toute évidence une ancienne version de *Calamus* qui comporte encore quelques bogues. La dernière version de *Calamus* est la 1.09N. Demandez une mise à jour (update) de la vôtre à l'éditeur du logiciel (**ALM**) par l'intermédiaire de votre revendeur.

Le tableur K-Spread 4: un bon substitut d'Excel 3 sur ST.

Qui commercialise FMSGDOS? Qui commercialise NVDI?

FSMGDOS n'est pas encore disponible mais le sera très prochainement (sans doute avant la fin de l'année).
Ce produit sera édité par **Atari France** et commercialisé par tous les distributeurs Atari.
NVDI est, quant à lui, commercialisé par **Euromatique Technologie**.

Existe-t-il un soft équivalent à Excel3 sur ST?

Il n'existe pas de version ST d'*Excel 3*, mais des logiciels de la même nature comme *K-Spread 4* (**Arobace**) ou *Graal Calc* (**Editions Profil**).

Je souhaite savoir si le MEGA STE peut accepter les logiciels Datamat, Textomat, Calcomat, ainsi que leurs fichiers établis actuellement sur 1040 STF.

Datamat et *Textomat* ne fonctionnent pas sur MEGA STE car ils ont été prévus pour le *TOS 1.2*. En revanche, *Calcomat* (version 2) marche sans problème (y compris la récupération de vos anciens fichiers). En ce qui concerne *Textomat*, vous pouvez sauver vos fichiers au format ASCII et les récupérer sur n'importe quel traitement de texte fonctionnant sur MEGA STE.

Les fichiers Textomat créés sur 1040 STE pourront être récupérés sur n'importe quel traitement de texte fonctionnant sur MEGA STE à la condition d'être préalablement sauvés au format ASCII.

Comment poser vos questions

1) par minitel (3615 ATARI)
C'est rapide: réponse sous 48 heures (maximum) dans la plupart des cas.
Posez votre question dans la rubrique Questions/Réponses. Revenez consulter la rubrique dès le lendemain, une réponse devrait vous attendre.
2) par courrier ou télécopie
Ecrivez à notre adresse en mentionnant sur l'enveloppe «Atari Magazine, courrier des lecteurs» et notre adresse. Si votre courrier est choisi par la rédaction, vous trouverez une réponse dans ces colonnes dans un prochain numéro (délai de réponse entre 3 et 8 semaines).
CE QU'IL NE FAUT PAS FAIRE
- téléphoner pour poser directement la question à la rédaction,
- écrire en joignant (ou pas) une enveloppe timbrée pour une réponse individuelle,
- envoyer un fax pour obtenir une réponse par le même canal...
Nous ne sommes pas actuellement en mesure de donner satisfaction à ce genre de demande.

J'utilise, dans mon travail, *Works* et *Wordstar*. Est-il possible d'utiliser ces traitements de textes sur le 1040 STE de mon fils, et si oui comment?

Cela est complètement possible à condition que vous équipiez le 1040 STE de votre fils d'un émulateur PC. Celui qui conviendrait le mieux est sans doute l'Atonce 386SX de Vortex.

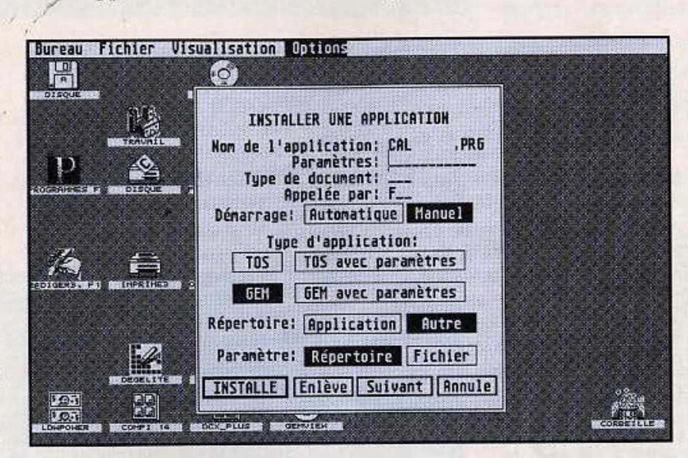
J'ai téléchargé LGSELECT.PRG (sélecteur d'objet) que j'ai installé avec l'option «Installer une application». Puis, pour l'option «Type de document» j'ai mis PRG. Pour l'option «Lancement» j'ai choisi «Normal» et pour «Type d'application» j'ai choisi «GEM». Depuis, l'ordinateur interdit tout lancement des programmes! Que faire?

Le programme *LGSelect* est un utilitaire qui doit être automatiquement chargé dans la mémoire lors du démarrage de votre ordinateur.

Pour cela, vous devez le copier dans le répertoire «AUTO» de votre unité de boot (disquette ou disque dur).

Après avoir effectué cette opération, il vous faudra résoudre votre problème de configuration qui empêche le lancement de vos programmes. Deux solutions:

- détruire le fichier `Desktop.inf`, mais vous perdrez du même coup les informations concernant votre bureau (position des fenêtres, nombres et position des icônes, etc.),
- refaire, à l'envers, l'opération d'installation que vous aviez effectuée: cliquez sur l'ancienne version de *LGSelect*; faites «Installer une application» puis rétablissez les choix (si vous ne vous en souvenez plus, inspirez-vous d'une application non encore installée).



Un exemple d'application non encore installée.

***Neochrome 224* a une documentation en anglais.**
Quelle est la référence de la documentation en français à télécharger?

Elle est disponible en téléchargement sous le nom `Neo224F.Tos`.

Existe-t-il un logiciel qui permet de s'entraîner à taper sur son clavier?

Il en existe un en téléchargement sur le 3615 ATARI.

Son nom est *Clavier*. Amusez-vous bien et attention aux cals au bout des doigts!

Devenez un professionnel du clavier.

INITIATION

Comment trifouiller l'intérieur d'un fichier?

Il vous faut un éditeur de secteur qui vous permettra de visualiser le contenu d'une disquette sous forme de caractères ASCII (texte) ou de codes hexadécimaux. Vous pourrez ainsi rechercher l'information qui vous intéresse dans un fichier, et la modifier à votre guise. De tels logiciels sont disponibles sur le 3615 ATARI, l'un des plus complets et des plus performants étant *STDoctor*.

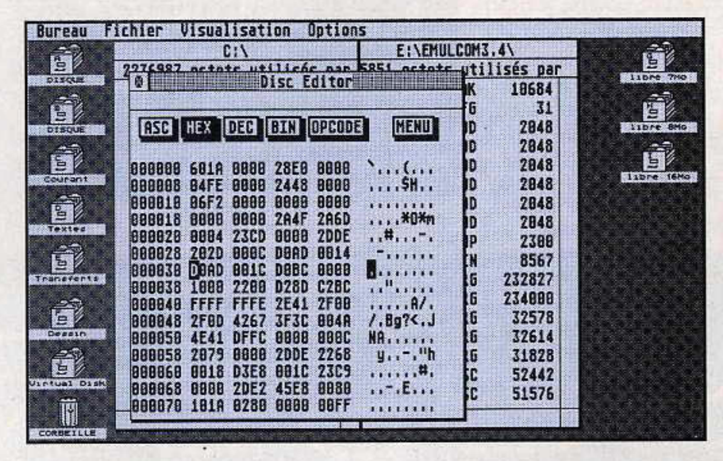
Dans le cas d'un fichier texte, un simple éditeur de texte vous suffira. Plusieurs sont disponibles en téléchargement. Pour plus de renseignement, reportez-vous à l'article qui leur est consacré dans le numéro 31 d'Atari Magazine.

Que veut dire ST?

ST signifie Seize Trente-deux bits (Sixteen Thirty two bits), car le microprocesseur de l'Atari (le MC 68000) dispose d'un bus externe de données de 16 bits (pour les échanges avec la RAM) et d'un bus interne de 32 bits (pour tout ce qui concerne les instructions en langage machine et les échanges de données avec les registres). Le ST fut le premier micro-ordinateur équipé d'un tel microprocesseur, ses prédécesseurs n'étant munis que de 16 bits.

Comment fait-on pour démarrer un programme au lancement de la machine avec un TOS 1.2?

Il vous faut le programme *Startgem* (disponible en téléchargement sur le 3615 ATARI) que vous installerez dans le répertoire Auto de votre disquette de démarrage, ou, si vous possédez un disque dur, sur la racine de la partition C de ce dernier. Puis vous devrez créer, à l'aide d'un éditeur de texte (ou d'un traitement de texte), un fichier ASCII (texte) contenant le nom du programme à exécuter au démarrage ainsi que son chemin d'accès. Ce fichier devra porter le nom `Startgem.inf` et se trouver sur la racine de votre unité de démarrage (disquette ou partition C du disque dur).



STDoctor est l'un des éditeurs de secteur les plus puissant du domaine public. L'intérieur de vos fichiers n'aura plus aucun secret pour vous.

Que signifie le message: «ERREUR TOS#35»?

Cela signifie que le système d'exploitation de votre ST est débordé et qu'il ne peut plus ouvrir de fichiers supplémentaires. En fait, ce problème arrive le plus souvent lors d'une mauvaise copie de fichier.

MATERIEL

Est-il possible de brancher un disque dur d'un autre type que SCSI (IDE par exemple)?

C'est tout à fait possible, mais il vous faudra utiliser la toute dernière version du driver pour disque dur Atari: *AHDI* (la version 5.0) disponible en exclusivité sur le 3615 ATARI.

Peut-on trouver un câble qui se branche sur l'interface parallèle du Portfolio et qui le relie au MEGA STE pour faire des transferts de fichiers.

Vous devez tout d'abord vous procurer un «câble parallèle» que vous trouverez chez n'importe quel bon revendeur informatique. Puis, pour pouvoir effectuer vos transferts de fichiers, il vous faudra utiliser le logiciel *FolioLink* vendu par la société **MC3**.

Nous possédons depuis dix ans un «Vidéo Computer System» de chez Atari, avec seulement des joysticks et très peu de cassettes pour jouer.

Nous voudrions compléter notre collection de jeux pour en faire profiter nos enfants.

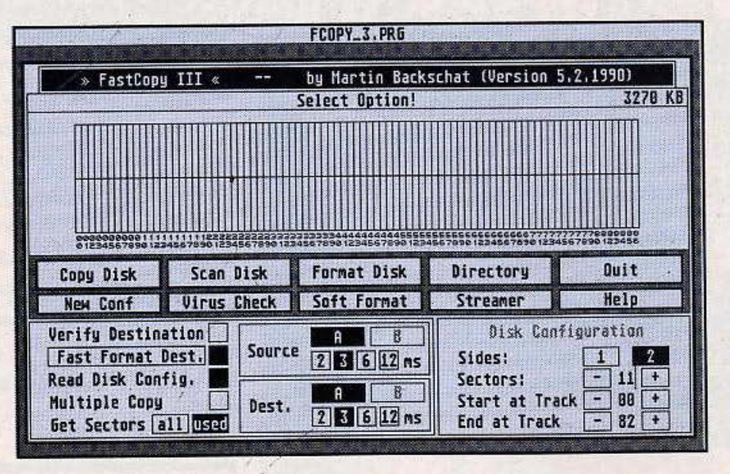
Pourriez-vous me renseigner car c'est un matériel qui ne se fait plus.

Solides les Atari! Rassurez-vous, vous pouvez utiliser les jeux destinés à la console 2600 d'Atari qui est entièrement compatible avec celle que vous possédez. Pour les jeux adressez-vous à **Innelec** ((1) 48 91 00 44) ou à **Guillemot** ((16) 99 08 83 54).

NOUVEAU
Dialoguer sur le 3615 ATARI

Un problème avec votre ST? Une bouffée de solitude vous envahit soudain?

Des animateurs d'Atari Magazine sont désormais à votre écoute sur le 3615 ATARI rubrique Dialogue (DIA), tous les jours de la semaine de 17 heures à 18 heures. N'hésitez pas à leur exposer tous vos soucis, ils se feront un plaisir de vous aider. Un lieu de rencontre sans équivalent est enfin à la disposition des Ataristes de la France entière. Echangez, en direct, vos idées et vos impressions sur les domaines qui vous passionnent. A ne louper sous aucun prétexte.



Augmentez la capacité de vos disquettes de 711 Ko utilisables à 903 Ko grâce au programme Fcopy 3.

Pour les périphériques (et les jeux), téléphonez à **APAK** ((1) 46 78 28 14).

DECOMPACTAGE

Comment faire pour décompacter Globe.arc. Arcx.ttp ne veut pas s'ouvrir, il m'indique «Paramètres» et ne veut rien accepter. Existe-t-il une procédure spéciale?

Quand le programme de décompactage `Arcx.ttp` vous indique «Paramètres», il attend que vous lui fournissiez le nom du fichier à décompacter (par exemple `Globe.arc`). De plus, `Arcx.ttp` doit être copié sur la disquette contenant le programme à décompacter ou, si vous disposez d'un disque dur, sur la même partition et dans le même répertoire.

Comment faire pour décompacter un programme dont la somme des fichiers décompactés a une taille supérieure à la capacité d'une disquette.

Il vous suffit d'étendre la capacité de votre disquette grâce au programme *FCopy3* (disponible en téléchargement): augmentez la valeur du nombre de secteurs par piste («Sectors»), ainsi que celle du nombre de pistes («End at Track») présentes en bas à droite de l'écran principal de *FCopy3*. Les valeurs maximales autorisées sont 11 pour la première et 82 pour la seconde. Ensuite, formatez votre disquette en cliquant sur «Format Disk», et vous obtiendrez une disquette de 903 Ko utilisables (au lieu de 711 Ko).

J'ai un problème en décompactant Sozobon. Le premier fichier Sozocl.arc passe sans problème, mais les deux suivants refusent de se décompacter. C'est frustrant...

C'est parce que *Sozoc2* et *Sozoc3* possède une extension `.ar` et doivent donc être décompactés avec `Archive.prg` et non pas avec `Arcx.ttp`. Adieu la frustration!

Je n'arrive pas à décompacter STAD 2, pourtant j'ai suivi vos directives en utilisant le programme Archive avec, sur la première ligne: A:\STAD12D.AR, et sur la deuxième ligne: A:\... A ce moment là, le décompactage commence bien, mais s'arrête rapidement. L'ordinateur marque alors «\n expected in Archive» puis «EXIT», et sur la disquette apparaissent deux dossiers: OB3STAD et STADFONT, mais ils sont vides! Que faire?**

Vous avez fait une erreur sur la deuxième ligne. Il ne faut pas taper `A:\.**.` mais `A:\*.*` pour que le programme *Archive* décompresse tous les fichiers (`*.*`) sur la disquette A (`A:\`).

3615 ATARI

Impossible de télécharger id.cpx, quel que soit le style d'écriture (majuscule ou minuscule). Pouvez-vous me donner la procédure exacte?

`id.cpx` s'appelle en fait *Vanity*. Ce dernier est un fichier autodécompactable (`.TOS`). C'est lui que vous devez télécharger. Après décompactage, vous retrouverez le fameux `id.cpx`.

Comment fait-on pour connaître le prix d'un programme téléchargé? Correspond-t-il à la durée de téléchargement?

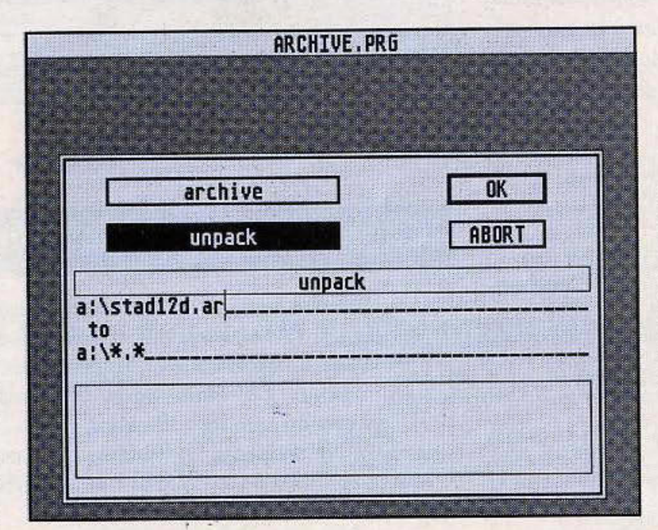
Le coût du téléchargement d'un programme sur le 3615 est effectivement fonction de la durée du transfert. Comme le prix d'une communication en 3615 est de 0,98 F par minute, multipliez ce chiffre par le temps que vous passez à télécharger un programme, et vous obtiendrez le coût exact de celui-ci. En moyenne, le téléchargement s'effectue à raison de 5 Ko à la minute.

Comment télécharger sur votre serveur avec mon MEGA STE? Avec mon MEGA ST et votre cordon cela fonctionnait très bien. Peut-on toujours utiliser ce cordon avec le MEGA STE, éventuellement avec un adaptateur 25 broches/9 broches et sur quel port du micro?

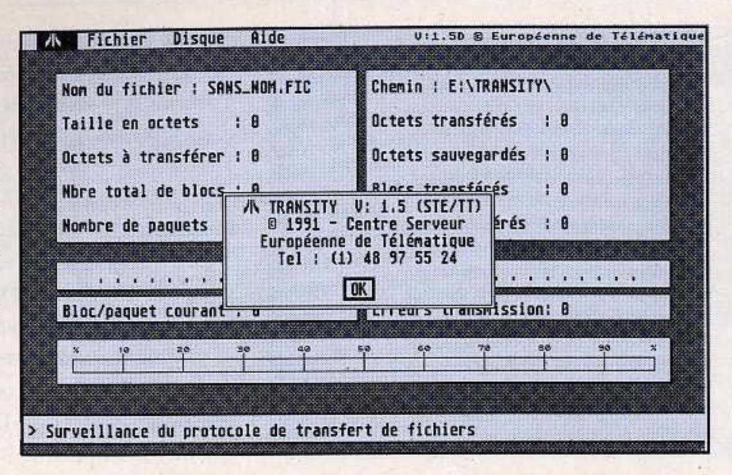
Vous pouvez bien sûr utiliser le même cordon et vous aurez effectivement besoin d'un convertisseur 25 broches/9 broches que vous devrez connecter sur le port modem n°1. Mais il vous faudra utiliser, au minimum, la version 1.5D de *Transity*.

Je possède un minitel 1 et je n'arrive pas à télécharger avec Transity que je viens d'acheter. Que dois-je faire?

Le minitel 1 n'est pas bi- standard (avec modem retournable). Pour télécharger avec *Transity* sur le 3615 ATARI vous devez donc échanger votre minitel 1 contre un minitel 1B ou un minitel 2.



«A:\» désigne le lecteur de disquette A, et «*.*» indique au programme Archive qu'il doit décompresser tous les fichiers se trouvant dans l'archive (le programme compressé).



La version 15D de Transity est au minimum nécessaire pour télécharger sur le 3615 ATARI avec un MEGA STE.

Je possède un modem Olitec 1200 (compatible Hayes), le logiciel *Emulcom 3*, le tout sur STE. Tout est bien relié et je n'arrive pourtant pas à télécharger. *Emulcom 3* est-il bien configuré?

Il faut configurer *Emulcom 3* avec le protocole CCETT pour pouvoir télécharger sur le 3615 ATARI. Le choix du protocole de transfert vous est demandé par *Emulcom* à chaque fois que vous désirez télécharger. Vous n'avez qu'à cliquer sur l'option CCETT.

J'ai un Portfolio et un programme de transfert ST/Portfolio. Mais en consultant votre liste des logiciels PC disponibles en téléchargement sur le 3615 ATARI, je ne sais pas si certains programmes fonctionnent sur le Portfolio. Si oui comment les reconnaître?

Les programmes qui fonctionnent parfaitement sur Portfolio sont ceux qui se trouvent dans le répertoire «Portfolio» du choix «TELST» (choix n°1) de la rubrique «Téléchargement».

Certains autres programmes pour PC disponibles sur le 3615 ATARI sont susceptibles de fonctionner sur Portfolio, à la condition de posséder une configuration adéquate (taille mémoire suffisante, lecteur de disquette PC, etc.).

Puis-je télécharger depuis l'île de la Réunion?

La Réunion c'est la France, donc pas de problème pour télécharger! Vous ferez nos amitiés à nos compatriotes d'outre-mer.

Comment obtenir les minuscules avec un Minitel 1B ou 2?

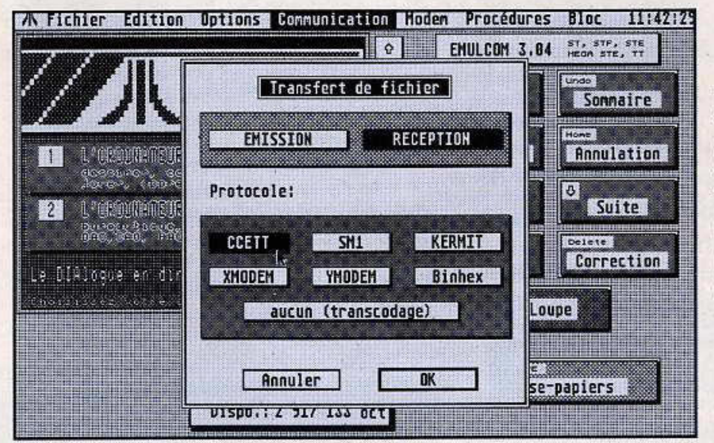
En tapant simultanément sur la touche [Fnct] et sur la touche [C], puis sur la touche [M]. Pour repasser en majuscule il vous suffira de refaire exactement la même chose.

Cependant sur le 3615 ATARI, il vous suffit de tapez `*MIN` puis la touche [Envoi] pour passer en mode minuscule, et `*maj` pour repasser en majuscule.

Je n'ai pas de Minitel. Existe-t-il un autre moyen de télécharger ou de se procurer les logiciels que vous proposez sur le 3615 ATARI?

Actuellement le seul moyen de se procurer ces logiciels est de les télécharger, ce qui représente le moyen le plus souple et le plus rapide d'étendre votre bibliothèque de programmes. En revanche, vous n'êtes pas obligé d'utiliser le minitel pour cela. Vous pouvez vous servir d'un modem répondant à la norme HAYES, auquel vous ajouterez le logiciel *Transity* proposé par **Atari Magazine** (voir bon de commande en fin de magazine). Cependant, un modem reste relativement cher en comparaison du minitel qui est fourni gratuitement par **France Télécom** ou proposé en location pour une somme relativement modique (20 F par mois pour le minitel 2).

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer les listings d'Atari Magazine n° 34. Pour vous les procurer, téléchargez-les sur le 3615 ATARI ou procurez-vous la disquette correspondante (voir bon de commande encarté à la fin du numéro).



Le téléchargement de programme sur le 3615 ATARI avec Emulcom 3 nécessite de configurer ce logiciel en mode CCETT.

Couleur haute définition sur MEGA STE

De nombreux lecteurs nous ont écrit à propos de notre dossier PAO d'Atari Magazine n°33 où nous disions: «Cependant, vous ne pourrez en aucun cas tirer parti à l'écran de la couleur haute définition, cette dernière n'étant accessible que sur une station TT». Nous tenons à préciser que cet avis ne s'appliquait qu'au MEGA STE en version standard. En effet, grâce à l'architecture ouverte de cet ordinateur, il est possible de lui adjoindre une carte couleur (Matrix, par exemple), qui vous fera alors bénéficier de la couleur haute définition à l'écran.

3615 JESSICO SUPER QUIZZ !
CONSOLES MEGADRIVE A GAGNER

JESSICO

"Quand les prix sont si bas, les souris dansent"

COMPILATIONS

THE BITMAP BROTHERS 252
Xenon + Cadaver + Speedball 2
AWARD WINNERS 292
Space ace+Kick off 2+Pipemania + Populus
SIMULATION'S BEST 295
Panza + Disk + ADS
ACTION KIT 265
Gemstone legend+Magicien+Builderl
TNT 2 249
Hydra + Skull & Crossbones + Badland + Targhan
ALBERTVILLE 1992 262
Game winter edition +Winter games + Superski
EXTRA BALL 269
Bumpy+Tennis+Sliders+Pinball M.
SIMULATION TOP 299
Panza K+Prince de Perse+Targhan
LES DIEUX DU SPORT 249
Tennis+SuperSki+Harricana+Outboard
MOVIES STARS 299
Dick tracy+Moonwalker+Indiana jone
SOCCER STARS 299
E.Hughes intern.soccer +Gazza 2 Kick off 2 + Microprose soccer
SUPER HEROES 299
Last ninja 2 + L espion qui m aimait Indiana jones crusade +Strider 2
LA COMPIL INTEGRALE 299
Lotus turbo+Toyota celica +Combo racer +Team Suzuki
DELTA FORCE 299
Fire forget 2 + Barbarian 2+ Knight force +Dark century
STARS SIX 299
Fire forget 2 +Mystical + Dark century +Swap+Crazy cars 2+off shore warrio
LES COLLECTORS 285
Lotus 1+Venus+J.pond+ Ghouls n'gh
5 INTELLIGENT GAMES 285
Backgammon + Chess + Bridge + Go +Draughts
SUPER ACTION 299
Supercars +Impossamole +Toyota ce G.Lineker + Switchblade
MAITRES AVENTURES 342
Maupiti +Voyageurs temps+O.stealth
SWORD & MAGIC 249
Targhan+Barbarian 2+Crystal arbore
TOP 3 289
Tennis + Pinball magic+Moonwalker
FOOTBALL CRAZY 285
Kickoff 2+Player manager+Final whis
SUPERSTARS SPORT 299
Beach volley+Rugby simulator Great court + Kick off
3 D SUPERSTARS 299
F29+Hard drivin+Stuncar+Battle com
SUPERSTARS DE L ARCADE 299
Golden axe + Narc + Super off road racer + Midnight resistance
NRJ 3 299
F16 +D.dragon 2+Italy 90+Welltris
ACTION PACK 265
Turrican 1+2+ Kick off + X-out
LES JUSTICIERS 3 299
Batman+ Robocop2+ Shadow warrior
OCEAN ARCADE 299
Midnight resistance + Golden axe +Off road racer + Narc
OCEAN 3D 299
Battle command + F29 + Hard driving
OCEAN SPORT 299
Manchester u.+ Pro tennis tour Beach volley + Off road racer
AVENTURES EXTRAORDIN. 295
Zac mc cracken + Iron lord Rocket ranger + Manoir morteville
AIR COMBAT ACES 325
Falcon + Gunship + Bomber
LES BATTANTS 2 285
Rick dangerous 1&2 +Satan Licence to kill + Double dragon 2
10 MEGA HITS 3 349
Stunt car racer + Foot man 2 Last ninja 2 + Highway 2+ Rank 3 slooges + Trivial 2 +Apb + Defender + Tetris
AIR SEA SUPREMACY 285
Silent service +F15 +P47+Carrier command +Gunship
KARATE ACES 295
Double dragon 1&2 + Saboteur 2 +Last ninja 2 +Oriental games
QUEST AND GLORY 299
Cadaver + Midwinter + Iron lord +Bloodwych
10 GREATS GAMES 332
Ferrari f1 +R.dangerous+Pick'n pile +Night hunter+Carrier command +Tintin+Super ski +Chigago 90 +Xenon 2 +Pro tennis tour
TOP LEAGUE 299
Speedball 2 Rick 2 +Falcon Midwinter +Tv sport footbal
FUN RADIO 299
Tortues ninja + Gremlins 2+Day of thunder + Back to the future 3
TOP ACTION 225
Hard drivin + Strike f.harrier Pic'n pile +Licence to kill
SUPER SEGA VOL 1 285
Shinobi + Super monaco gp Golden axe + Chuckdown +E-swat
CAPCOM COLLECTION 265
Strider 1&2 +Un squadron+Duel +Forgotten world +Ghouls'n ghots +Dinasty+Led storm
LE TEMPS DES HEROS 289
North & south + Moonblaster + Prince of persia
LES STARS 289
Skweek +Superskweek+Bumpy Builderland
SUCCES STORY 229
Bumpy + Skweek + Turbocup crazy Shot + Space racer
PLANETE AVENTURE 292
Indi. jones a.+Maniac mansion Explora 2+Portes du temps
MEGAMIX 279
Turrican + Chase hq +Shadow warriors + Altered beast
LES GEANTS DU SPORT 349
Great courts+Kick off +Foot. manag.2
STARS D'HOLLYWOOD 249
Batman+Indiana jones+Robocop+ Ghostbusters 2

LOGICIELS JEUX

3D CONSTRUCTION KIT 375
ADS + SHERMAN 282
AIGLE D OR" LE RETOUR" 282
AIRBUS A320 342
AIR SUPPORT 252
AIR WARRIOR 342
ALCATRAZ 242
ALIEN STORM 252
ANOTHER WORLD 292
AVENTURES DE MOKTAR 295
BABY JO 282
BARBARIAN 2 252
BARON ATTACK 292
BAT 2 342
BATTLE ISLE 292
BATTLE OF BRITAIN 292
BEAST BUSTERS 252
BETRAYAL 292
BIRDS OF PREY 342
BLACK CRYPT 252
BOMBER MAN 292
BONANZA BROS 252
BOSTON BOM CLUB 252
CASTLES 292
CHAMPIONSHIP MANAGER 242
CHAOS ENGINE 252
CONAN 292
COVERT ACTION 342
CRAZY CARS 3 252
CRIME CITY 292
CROISIERE P. UN CADAVRE 272
DAEMONGASTE 342
DAS BOOT 292
DEMONIAC 292
DEUTEROS 292
DICK TRACY AVENT 302
DIE HARD 2 252
DIPLOMACY 195
D. GENERATION 302
DUCK TALES 252
DUNE 342
DYNABLASTER 305
ELVIRA 2 342
EPIC 252
EYE OF THE STORM 292
F15 STRIKE EAGLE 2 342
FANTASY PAK 192
FIGHTER COMMAND 292
FINAL BLOW 252
FINAL FIGHT 252
FIREBALL 252
First samurai + Megalomania 292
FLAG 292
FUTURE DREAMS 262
GEM'X 262
GLOBAL EFFECT 342
G.LOC 252
GUY SPY 299
HARLEQUIN 252
HEIMDALL 292
HERO QUEST + DATA DISK 242
HOOK 252
HUNTER 292
INDY HEAT 252
ISHAR 285
JAGUAR XJ 220 292
KAWASAKI 262
KILLERBALL 292
KNIGHTMARE 292
KNIGHTOF LEGEND 299
KNIGHT OF THE SKY 342
KICK OFF 2 + SCENARIO 249
KO 2 FINAL WHISTLE 125
KO 2 GIANTS OF EUROP 129
KO 2 RETURN TO EUROP 125
KID GLOVES 2 252
LAST NINJA 3 252
LEANDER 292
LEGEND 252
LIVERPOOL 252
LOTUS TURBO CHALLENGE 2 252
M1 TANK PLATOON 292
MAGIC POCKETS 252
MANCHESTER UNITED EURO 252
MEGA TWINS 252
MERCENARY 3 292
MICKEY CROSSWORD 252
MICROPROSE GOLF 342
MICROPROSE GRAND PRIX 342
MIG 29 SUPER FULCRUM 392
MONKEY ISLAND 2 292
MONSTER PACK 2 252
MOONFALL 252
MYTH 252
NASKAR 342
OBITUS 252
OMAR SHARIF BRIDGE 342
ORK 252
PACIFIC ISLANDS 292
PANZA KICK BOXING 282
PAPER BOY 2 292
PARASOL STAR 252
PITFIGHTER 2 252
PLAN 9 252
POPULUS 2 292
REALMS 292
REACH FOR THE SKIES 302
RIDERS OF ROHAN 252
ROBOCOP 3 252
ROBOZONE 252

LOGICIELS JEUX

RODLAND 252
ROGER RABBIT 252
ROLLING RONNY 252
ROTATOR 252
RUGBY THE WORLD CUP 242
RUGBY TOURNOI 5 NATIONS 292
SHADOWLANDS 252
SHADOW OF THE BEAST 2 252
SHOCK WAVE 252
SHUTTLE 442
SILENT SERVICE 2 342
SIMANT 292
SIM EARTH 252
SIM CITY + POPULUS 292
SMASH TV 252
SNOW BROSS 252
SPACE CRUSADE 252
SPACE ACE 2 392
SPACE GUN 252
SPECIAL FORCES 342
SPEEDBALL 2 252
SPIRIT OF EXCALIBUR 292
SPOT 192
STARUSH 242
STEEL EMPIRE 292
STORM ACCROSS EUROPE 299
STORMBALL 252
STORM MASTER 292
STRIKEFLEET 252
SUMMER CAMP 252
SUPER SKI 2 292
SUSPICIOUS CARGO 252
SWORD OF SAMOURAI 292
TERMINATOR 2 252
THE ADDAMS FAMILY 252
THE BLUE MAX 292
THE MANAGER 252
THE SIMPSONS 252
THE SHOE PEOPLE 252
THUNDERHAWK AH-73M 292
TWILIGHT 2000 342
TYPHON 265
TILT 255
TOP BANANA 252
TORTUES NINJA 2 252
TURBO CHARGE 272
ULTIMA 6 305
UTOPIA 292
UTOPIA DATA DISK 145
VIDEO KID 252
VROOM 252
VROOM DATA DISK 149
WALKER 252
WARLORDS 252
WARLOCK THE AVENGER 252
WARZONE 252
WINTER SUPERSPORTS 92 292
WIZKID 252
WOLFCHILD 272
WOLF PACK 252
WONDERLAND 292
WORLD CLASS RUGBY 292
WRECKERS 252
W.W.F 252

JOYSTICKS

COMPETITION PRO 149
PRO 5000 149
QUICKJOY JUNIOR 59
QUICKJOY TURBO 99
QUICKJOY 2 89
QUICKJOY 3 SUPERcharger 109
QUICKJOY 5 SUPERBOARD 199
QUICKJOY VI JETFIGHTER 149
QUICKJOY TOPSTAR 295
QUICKJOY NI-5 Nintendo 159
QUICKJOY SEGA FIGHTER 169
QUICKJOY Turbo Pedale 279
QUICKJOY Megastar Jùnior 199
QUICKJOY MEGASTAR 329
QUICKJOY SUPERSTAR 179
QUICKJOY TOPSTAR PC 299
CARTE JOYSTICK -> PC 159
KONIX SPEEDKING 109
KONIX SPEEDKING Autofire 129
KONIX THE NAVIGATOR 149
STING-RAY 149
QUICKSHOT MAVERICK 1 169
QUICKSHOT PYTHON 1 119
QUICKSHOT INTRUDER 1 425
QUICKSHOT AVIATOR 1 475
QUICKSHOT FLIGHTGRIP 1 149

RUBANS

CITIZEN 120D 59
CITIZEN SWIFT 24/9 59
EPSON LX 800/LQ 800 59
ATARI SMM 804 59
STAR NL 10 59
STAR LC 10/20 59
STAR LC 10 COULEUR 99

TOUS LES RUBANS NB PAR 3 =144

HOUSSES

COMPLETE CLAVIER+ECRAN
CITIZEN 120 D 79
STAR NL 10 79
STAR LC 10 79
ATARI ST 119
SMM 804 99

BOITIERS

AVEC SERRURE+CLÉ
BOITIER DD 50L : 50X5.25" 119
BOITIER DD 100L : 100X5.25" 149
BOITIER DS 40LB : 40X3.50" 99
BOITIER JSY 80 : 80X3.50" 109

UTILITAIRES

A.DEBOG 590
ASTROCYCLE SENIOR MONO 990
AUTOFORM.BASIC GFA 285
AUTOFORM.STOS BASIC 275
BECKERCAD DAO 975
BECKERCALC 975
BECKERTEXT II 725
BUROTEXT 775
CALCOMAT II 575
CLAVISSIMO 280
COPIEUR SYNCHRO EXPRESS 395
CYBER PAINT 2.0 690
DALI 4 575
DEGAS ELITE 245
DELUXE PAINT 485
DEVPAC V.2 730
DISCOSCOPIE 250
FACT+STOCK FACILES 395
FOLIOLINK ST 590
GEM UP 289
GESTCOMPTE 2 329
GESTCOMPTE 2 + 3614 459
GESTION BUDGET Personnel 300
GFA ASSEMBLEUR 575
GFA BASIC 3.0 690
GFA COMPILATEUR 3.0 325
HABA TRAITEMENT TEXTE 895
LE COMPTABLE 2 775
LE DESSIN TECHNIQUE 1490
LE DESSINATEUR 590
LE GESTIONNAIRE 590
LE PROFESSEUR 790
LE REDACTEUR 3.15 990
L 'ELECTRONICIEN 975
LE TAROMANCIEN 290
LOTO 590
LOTO EXPERT 890
MEGA FILECOPY 185
MEGA STICK 185
MISE A JOUR GFA 3.5 275
MUSIC MASTER + CARTE 425
MUSIC ONE 375
PACK ANTIVIRUS 199
PACK GFA BASIC 2.0 190
PACK GFA BASIC 3.0 775
PAINT MASTER 590
PERMIS CONDUIRE CIRCUIT D 249
PERMIS CONDUIRE PANNEAUX 249
TABLEAU DE BORD D 249
TEST PERMIS CONDUIRE 1 D 249
TEST PERMIS CONDUIRE 2 D 249
PREVISIONS ASTRALES 390
PRO 12 575
RACINES (GENEALOGIE) 390
SPRITE EDITOR DELUXE 250
ST BUDGET 390
ST COMPTES 290
ST REPLAY V.8 690
STARTER PACK 580
STOS 3D 395
STOS BASIC 395
STOS COMPILER 269
STOS MAESTRO 299
STOS SPRITES 199
THE HMS SOUNDTRACKER 390
ULTIMATE RIPPER 490
WERCS 325

LIBRAIRIE

AUTOFORMATION BASIC + D 199
BIEN DEBUTER ATARI 89
BIEN DEBUTER GFA B.2-3-3.5 129
BIEN DEBUTER STOS 129
BIEN DEB TEXT.DAT.CALC.+ D 199
DEVELOP.SUPERBASEPRO+D 299
DISQUET ET DISQUE DUR + D 279
GRAND LIVRE DU ST + 2D 199
LIVRE ST WORD + ... 165
LIVRE DELUXE PAINT 129
LIVRE DEVELOPPEUR 1 199
LIVRE DEVELOPPEUR 2 199
LIVRE DEVELOPPEUR + 2D 299
LIVRE GFA BASIC 3-3.5 199
LIVRE GFA BASIC 3-3.5 +D 265
LIVRE DES IMPRIMANTES 249
LIVRE DE L'OMIKRON BASIC 165
LIVRE LANGAGE MACHINE 149
LIVRE SUPERBASEII+PRO+III 169
LIVRE DU REDACTEUR 149
MIDI MUSIC ET SEQUENCEUR 99
PROGRAM GFA B.+ASSEMB+D 195
TOS 1.4 ET TOS STE 99

ACCESSOIRES

FILTRE ECRAN 14" 169
SOURIS GENIUS ST 249
LECTEUR INTERNE 3.50" ST 590
LECT.EXTERNE ALIM.220V 649
LECT.EXTERNE port joystick 549
STE SIMMS 1 MEG 410
EXTENS. STE 4 MEG-SIMMS 2390
CABLE PERITEL ST 139
ETIQUETTES 89X23 LES 500 69
ETIQUETTES 3.50" LES 100 32

SUPER PROMO

PACK GFA + Compil + GFA assembleur 775
STARTER PACK (datamat+textomat+calcomat+dompub +livre meilleurs jeux st+autoformation basic livre st) 580
PACK GFA BASIC 2.0 (interpreteur GFA basic 2.0 compilateur GFA +bien debuter GFA 2.0) 190
PACK GFA BASIC 3.5E (INTERPRETEUR + compilateur) 975
IMPRIMANTE STAR LC200 Couleurs 2390

COPIEUR SYNCHRO EXPRESS 2

- Copie des disquettes en moins de 30 secondes.
- SYNCHRO EXPRESS nécessite un 2éme lecteur.
- Sélection par menu des débuts et fins de pistes (jusqu'à 85 pistes), une ou deux faces, autosélect.
- Duplication d'autres formats tels IBM, MAC, etc

PROBABLEMENT LE MEILLEUR COPIEUR !
SYNCHRO EXPRESS ST 395 F



DIGITALISEUR VIDI N°1 EN EUROPE!

"LA RÉFÉRENCE " ..... **ATARI Magazine**

- DIGITALISE AU 1/50S TOUTE IMAGE DE SOURCE VIDEO - 32 K MEMOIRE,COMPATIBLE NEO / DEGAS - RESOLUTION 320 X 200 4 BITS PAR PIXEL- CONTROLE DU CONTRASTE ET LUMINOSITE

PACK VIDI NEW ST OU AMIGA 1290 F
ce pack comprend le digitaliseur **VIDI** + **VidiChrome** + **PhotonPaint** (pour AMIGA) ou **Spectrum 512** (pour ST)

VIDI RGB FILTRES COULEURS ELECTRONIQUES 799 F

PACK VIDI COMPLET ST OU AMIGA 1790 F
COMPREND **PACK VIDI NEW + VIDI RGB**

RÉF.	par 10	par 20	par 50
3" 1/2 DFDD	69 F	130 F	300 F
5" 1/4 DFDD	56 F	105 F	250 F

DISQUETTES 3.50" DFDD

DISQUETTES CERTIFIÉES 100%, GARANTIE À VIE LIVRÉES AVEC ENVELOPPES + ÉTIQUETTES . NEUTRES (VERBATIM)

DISQUETTE NETTOYAGE 3.50" : 59 F

NOUS LIVRONS TOUJOURS LES DERNIERES VERSIONS

BON DE COMMANDE *EXPRESS*

GAGNEZ DU TEMPS ! Commandez par ☎ 93.51.61.30 - 93.97.22.00

à retourner à **JESSICO** - BP693 - 06012 NICE CEDEX 1

TITRES (garantie échange immédiat)	Qte	Prix	Montant
		S/TOTAL	
		PORT	2 5
		TOTAL	

PORT : LOGICIEL JEUX 20 F
IMPRIMANTES + CONSOLES 60 F
UTILITAIRES + ACCESSOIRES 25 F
DOM TOM + ETRANGER + 60 F

PAIEMENT ETRANGER EXCLUSIVEMENT PAR MANDAT INTERNATIONAL

GARANTIE 1 AN SUR LES LOGICIELS ■ précisez votre ordinateur ______ DISC ☐ K7 ☐ ■ Votre N° Client ______

PAR MINITEL 3615 CODE JESSICO - OUVERT 7 / 7 - 8H à 20 H

☐ Je joins un chèque ou mandat lettre 9-05
☐ Je paie à réception au facteur + 25 F
☐ Je paie par carte bleue et je complète les 2 lignes ci-dessous

carte bleue CB

date d'expiration ....

NOM .... PRENOM ....
N° ET RUE ....
VILLE .... CODE POSTAL

SIGNATURE OBLIGATOIRE

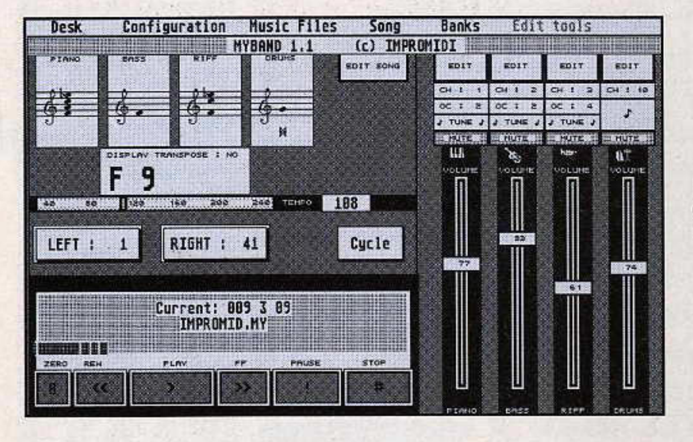
FAX : 93.97.07.00

BOUTIQUE A NICE HOLLYWOOD STAR 8 BD JOSEPH GARNIER

MYBAND
L'arrangement simplifié (version 1.1)

Ne soyez plus jamais un musicien solitaire face au STE. Quelque soit votre niveau, Myband met un quartet à la disposition de vos envies musicales les plus folles.

De nombreux amateurs de musique ont un jour souhaité pouvoir disposer d'un orchestre à la maison.
Basse, piano, cuivres, batterie, en somme un quartet docile pour les accompagner dans l'étude de leurs morceaux préférés.
Myband transformera votre rêve en réalité, car avec une interface graphique très musicale, il crée vos propres accompagnements dans tous les styles musicaux.
Le principe de *Myband* fait appel aux patterns, c'est-à-dire aux motifs musicaux d'une ou plusieurs mesures.
Livré avec une banque très complète de motifs musicaux de tout style, il vous suffit de les assembler dans le bon ordre selon une grille musicale déterminée.
Cette dernière peut être relevée sur les partitions du commerce, pour que le Top 50, le standard de jazz, ou le dernier succès à la mode soient joués par votre STE.
Pas besoin d'être un virtuose, *Myband* interprète les grilles d'accords en arrangements plus ou moins sophistiqués, envoyés sous forme de données MIDI au module de production sonore, synthétiseur, piano numérique, expandeur, etc.
Myband ne fonctionne pas avec le synthétiseur interne du STE.



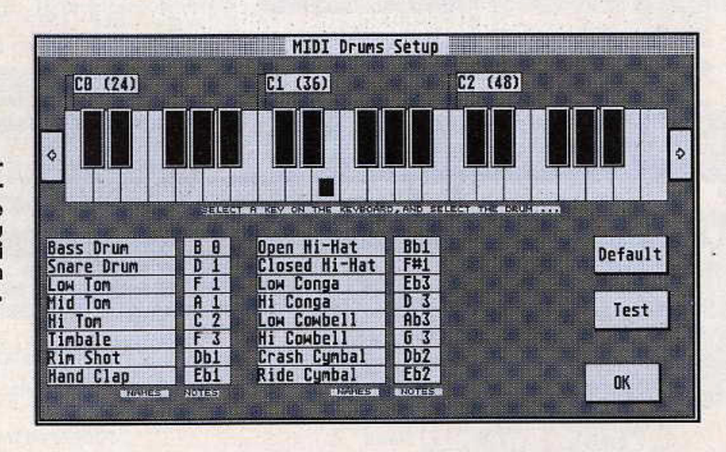
L'écran principal, avec son pupitre de mixage.

La configuration sonore

Pour le débutant, une des plus grosses difficultés réside dans la configuration du synthétiseur ou de l'expandeur, afin d'obtenir les sons désirés. Ici, nous avons quatre instruments à configurer: le piano, la basse, l'accompagnement et la batterie. Ce sont les quatre éléments de base d'un petit groupe de musiciens, et donc d'un arrangement. Le synthétiseur, ou l'expandeur (synthétiseur sans clavier), doit être multitimbral avec, au minimum, 4 voies pour reproduire les 4 parties de l'arrangement final. Il doit posséder au moins 24 notes de polyphonie réparties chacune entre les différentes voies. La quatrième voie, la rythmique, est une voie spécifique où chaque note correspond à un son de percussion. C'est ce qu'on appelle le «Drum Set» ou le «Rhythm Kit». C'est elle qui servira à produire la batterie. De nombreux synthétiseurs ou expandeurs, surtout les plus récents, sont conformes à ces caractéristiques, ayant adoptés le nouveau standard General MIDI, compatible avec le General Standard de **Roland**.

Des réglages simples

Myband optimise les réglages sonores grâce à un petit pupitre de mixage, comportant quatre potentiomètres et des options qui règle facilement le canal MIDI et l'octave correspondant à chacune des parties. La boîte «Tune» teste le son correspondant, la boîte «Mute» coupe le son de la voie pour isoler les autres voies comme sur une console professionnelle. Il vous suffit de régler le potentiomètre grâce à la souris pour faire varier le volume, rien de compliqué.
Le séquenceur intégré à *Myband* charge une des six démos (.MY). C'est un bon moyen d'apprécier la dynamique du logiciel, car pendant le jeu, vous voyez les notes corres-



Pour configurer la rythmique sur votre synthétiseur ou expandeur, voici un écran bien pratique.

L'éditeur de morceaux.

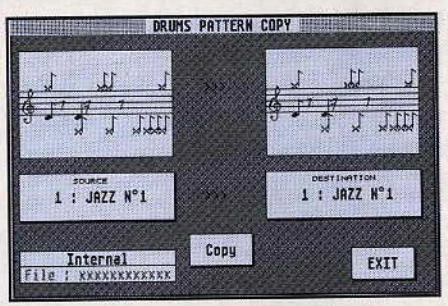
pondantes s'afficher dans les 4 fenêtres de portée (Piano, Bass, Riff, Drum) alors que les accords sont affichés en gros caractères pour que vous jouiez ou chantiez à vue. De plus, l'affichage des accords peut être automatiquement transposé en Si Bémol et Mi Bémol pour les sax ténor ou soprano qui veulent s'entraîner sur des grilles jouées par *Myband*.

Pendant le jeu, les options du pupitre de réglage restent actives. Le démarrage et l'arrêt du séquenceur sont commandés par la barre d'espace ainsi que par la touche [0] ou [Enter], ce qui est très pratique. L'avancement du morceau est visualisé par mesure jouée. Le séquenceur permet la remise à zéro, l'avance et le retour rapides, la pause, le réglage du tempo. Le jeu en boucle entre deux marqueurs donne le droit de travailler en boucle ou non sur un ensemble de mesures défini.

L'éditeur de morceaux

C'est dans cet écran graphique que vous entrez les grilles harmoniques définissant la structure du morceau.

Les patterns sont affichés sous forme de 4 petites fenêtres représentant des mini partitions. Cliquer dessus provoque le jeu du riff correspondant, de l'accompagnement, du



La copie graphique: c'est très simple.

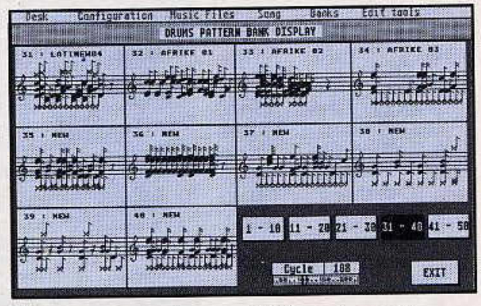
piano ou de la mesure de rythmique. Les quatre fenêtres peuvent être jouées simultanément, afin de juger de la qualité de l'ensemble instrumental et aussi des problèmes de polyphonie pouvant se poser aussi bien au niveau harmonique, qu'au niveau des limites du synthé ou de l'expandeur employé! La partie droite de l'écran, à l'allure de liste du type *Notator*, vous permet de rentrer la grille harmonique du morceau, avec les paramètres suivants: le numéro de la mesure, le nom de l'accord, le numéro de pattern de piano, celui de basse, le riff, la batterie et la durée de la mesure. Tous ces paramètres sont modifiables à la souris, peuvent être écoutés seuls ou globalement, en boucle ou linéairement, en fonction d'un tempo réglable dans cet écran.

L'éditeur autorise la copie, le déplacement de mesure, l'effacement, grâce aux boîtes «Bars Motion» et «Data Modification» du menu Edit Tools.

L'édition des patterns

Les patterns (motifs rythmiques, d'accompagnement, de basses, ou même soli) édités séparément, sont sauvegardés dans des banques spécifiques, afin de former des librairies pouvant devenir de vrais trésors d'arrangements et de rythmiques. Pour cela, quatre éditeurs de patterns sont disponibles, accessibles depuis l'écran principal ou par l'intermédiaire du menu Banks.

Le premier est l'éditeur de rythmique. Il fonctionne sur le principe de la grille d'édition. Cette dernière représente une mesure à 4 temps dans laquelle vous positionnez à la souris les 16 instruments de percussion correspondant aux notes définies. Vous avez la possibilité d'attribuer à ces notes 6 valeurs de vélocité. La partition musicale correspondante est affichée en bas de l'écran, sur une portée bénéficiant d'un affichage en temps réel, avec un tempo modifiable sur ce même écran. Toute la programmation fonctionne



Cinquante rythmes différents sont mémorisables.

Les arrangeurs sur ST

Il existe 3 types de logiciels d'arrangement sur STE. Premier type, c'est l'arrangeur interactif: vous avez un clavier MIDI, vous le reliez à votre STE, vous jouez sur le clavier et l'arrangeur s'occupe de tout. Il reconnaît et analyse les accords joués, compose l'arrangement et l'orchestration en temps réel. Feeling Partner de JCD/MPI est le parfait exemple de ce type d'arrangeur plutôt rare (voir Atari Magazine n°24). Le deuxième type est l'arrangeur en temps non-réel: vous entrez une suite d'accords, correspondant à une grille harmonique, et le type de rythme désiré, l'arrangeur va calculer l'arrangement musical à partir de ces données. Band in a Box est un bon logiciel qui a un très gros succès en Allemagne et aux Etats-Unis. (voir Atari Magazine n°29). Il existe de nombreux autres arrangeurs de ce type. Troisième type, l'arrangeur pédagogique qui fonctionne sur un principe de modélisation, illustré par Myband. Tout est programmable, la musique est créée comme un jeu de construction, à partir de petits motifs appelés Patterns. Il suffit ensuite d'assembler ces motifs en fonction d'une grille d'accord que l'on rentre comme dans Band in a Box. Moins automatique mais beaucoup plus créatif et très musical, ce type d'arrangeur est parfois considéré comme un outil de base à l'apprentissage de l'harmonie. En revanche, il demande un peu plus de connaissances musicales. D'autres arrangeurs, comme Mister Jam de la société Rythm'n Soft proposent des variantes, que nous analyserons prochainement.

Un écran pratique pour créer les riffs de cuivre.

sur ce principe de temps réel et de boucle, que ce soit pour la programmation des patterns de piano ou pour les accompagnements d'instruments aussi divers que les guitares ou les cuivres. La vélocité (nuances) attribuée aux accords, leur position temporelle par rapport au battement, ainsi que leur durée, sont programmées en écoutant continuellement le résultat en temps réel. En effet, le séquenceur incorporé fait tourner le pattern en boucle, ce qui correspond à la meilleure programmation interactive, si chère au musicien et à l'artiste.

Sur ce type d'écran, vous réalisez aussi des patterns d'arpèges devenant à leur tour des intros ou des solos. Toutes ces capacités ont pour seule limite l'imagination et le talent de l'utilisateur.

Mon orchestre à moi

Myband a été programmé par Christian Darcemont, avec la participation, en tant que conseiller musical, de Gérard Ramos.

Ce logiciel séquenceur, dédié à la création rapide d'accompagnements ou d'arrangements, s'adresse à tout musicien voulant programmer ses propres arrangements, avec des motifs préprogrammés. Il peut tout aussi bien les créer lui-même en s'inspirant des partitions et recueils divers.

Pour débuter, une petite formation théorique musicale et harmonique est nécessaire, car tous les accords sont au standard américain: A correspond au La, C à Do, et Am7 à La mineur Septième. Malgré un manuel en français de 92 pages, une version entièrement francisée du logiciel serait la bienvenue. En revanche, l'initié sera tout à fait à l'aise avec ce logiciel qui fait appel aux connaissances harmoniques de base d'un musicien.

Myband prend là toute son ampleur didactique, car avec un graphisme extrêmement bien réalisé (il effectue automatiquement la transcription des patterns en écriture musicale traditionnelle en plus de l'écoute temps réel), le musicien pourra tester, inventer, recopier des motifs rythmiques, des arrangements, les sauvegarder, se constituer une bibliothèque de divers motifs.

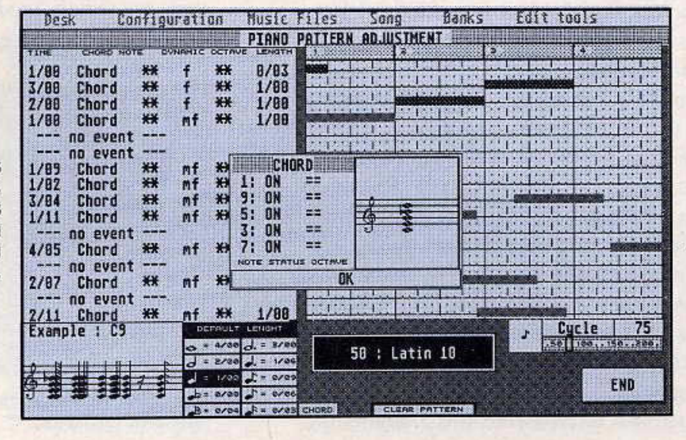
On peut effectuer une sauvegarde du morceau final en fichier standard Midifile, ce qui permet de le recharger dans un séquenceur professionnel tel *Notator*, *Cubase*, *Master Track Pro*, pour ne citer que les plus célèbres.

Programmez et testez les courbes de vélocité.

Toutes les manœuvres de programmation restent très compréhensibles pour le musicien, qui peut tirer pleinement parti de ses connaissances musicales, tout en les mettant facilement en application.

Al Jollyson ■

La grille d'édition de la rythmique.



Programmez les accords de piano, les notes d'arpèges et mettez-les en rythme avec la grille d'édition.

MYBAND
Distribué par **Impromidi**
Pour Atari ST/STE et TT.
Ecran monochrome ou couleur.
Affichage des accords transposables en Sib et MIb
Sauvegarde en Midifile. Quatre types de patterns (Piano, Basse, Riff, Drum).
200 Patterns maximum, banque de patterns fournie.
Prix: 950 F.

MAO III
Les messages MIDI

Les synthétiseurs, les boîtes à rythmes, les échantillonneurs... et bien sûr le STE, utilisent un langage standard commun pour communiquer et jouer de la musique ensemble.

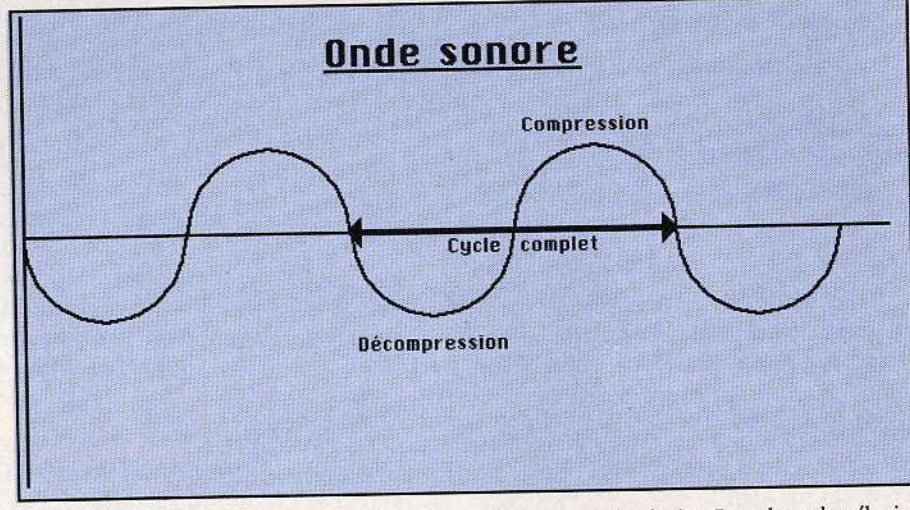
Nous avons vu précédemment que le MIDI était un standard hard, c'est-à-dire une norme physique (la prise, le branchement, le réseau...).
Mais le MIDI c'est aussi un langage de communication entre les instruments de musique pour qu'ils puissent jouer ensemble, quelque soit leur marque. Ce protocole est divisé en trois grandes familles:
- les «channel messages» qui indiquent approximativement les notes à jouer, la nuance, le son, le synthétiseur,
- les «messages système» pour la synchronisation, c'est-à-dire le moment auquel les notes sont jouées,
- les «system exclusive messages» pour le contrôle et la gestion du timbre du synthétiseur.

Channel message

Les «channel message» (en français «messages canal») sont des informations qui ne sont destinées qu'à un canal MIDI particulier (voir Atari Magazine n°33). La première sous-division que nous allons étudier se nomme les «channel voice messages» (messages musicaux de canal). C'est l'information la plus importante, car c'est grâce à elle que le morceau de musique est joué. Il existe sept types de «channel voice message» que nous découvrirons tout au long de cette initiation. Pour le moment, nous nous attarderons sur les plus importants.

Note on

C'est le message qui permet de déclencher une note sur un appareil MIDI. Il est émis par le clavier maître quand une touche est enfoncée. Lorsque ce message est reçu par le générateur sonore, ce dernier joue la note demandée.



Le message «note on» est composé de trois octets. Message de type:
1001cccc 0nnnnnnn 0vvvvvvv
Les quatre premiers bits du premier octet (1001) indiquent que le message reçu est une «note on» et qu'il faut faire attention aux deux octets suivants. Les quatre derniers bits du premier octet (cccc) indiquent sur quel canal MIDI le message doit aboutir. Il existe 16 canaux qui sont toujours codés sur les quatre derniers bits du premier octet dont la liste vous est donnée sur le tableau 1.
L'octet suivant (0nnnnnnn) indique la note jouée. Il commence toujours par un zéro. En MIDI, on a la possibilité de coder 127 notes, soit près de 11 octaves, ce qui est tout à fait suffisant, d'autant que la majorité des claviers grand public dépasse rarement 66 notes. La liste du codage des notes est indiquée en octet, en décimal et en hexadécimal dans le tableau 2 (pour les programmeurs).
Les notes sont réglées par rapport au La 3 dont la fréquence est de 440 Hertz (voir ci-contre).
Le troisième octet (0vvvvvvv), qui commence aussi par un zéro, indique la vélocité de la note, c'est-à-dire la force avec laquelle elle a été jouée sur le clavier. La valeur de vélocité peut varier entre 0 et 127 (de 00000000 à 01111111 en octet). Bien que cela soit de plus en plus rare, il arrive que certains claviers MIDI, surtout dans les bas de gamme et les orgues électroniques, n'émettent pas de vélocité. Ces types d'appareil envoient généralement comme code la valeur moyenne 64. Il sera ensuite toujours possible de modifier ces valeurs, sur l'écran du STE, grâce à un éditeur de séquenceur MIDI. Les claviers qui génèrent le code de vélocité s'appellent des claviers dynamiques. Il sont supérieurs, car ils permettent de jouer le morceau de musique avec nuance, comme sur un vrai piano. En réalité, ces claviers ne mesurent pas la force de la frappe sur la touche mais plutôt le temps mis par la touche pour s'enfoncer. Plus elle a été enfoncée rapidement, plus la vélocité est élevée, d'où une bonne restitution pour jouer en nuance sur un système MIDI.
Dans le générateur sonore de réception, la vélocité est généralement interprétée comme une augmentation de l'attaque du son pour que ce dernier soit plus fort. Mais il est tout à fait possible, selon les possibilités des ma-

chines, de créer d'autres effets. Par exemple, si l'on joue un Do 5 sur le clavier avec une force moyenne, le code émit sur le canal MIDI 2 sera 145, 84, 64 en décimal et en octet: 10010001, 01010100, 01000000.

Note off

C'est l'information complémentaire du «note on». En effet, une «note on» déclenche une note, mais ne l'arrête pas pour autant. Le clavier émet une «note off» quand la touche du clavier est relâchée. Après réception du message, le générateur sonore cesse de jouer la note. Bien sûr, ce message est ignoré si la note n'est plus jouée. C'est par le traitement de ces deux messages qu'il est possible de connaître la longueur de la note. A un tempo donné, réglé sur le séquenceur, le STE mesure le temps écoulé entre le message «note on» (début de la note) et le message «note off» (fin de la note). Si l'on joue à 4 temps, et que la distance entre les deux messages est de deux temps, la note sera une blanche. Si elle est de 1 temps la note sera une noire, etc.

La structure du message est la même que pour la «note on», c'est-à-dire trois octets.

Message de type:

1000cccc 0nnnnnnn 0vvvvvvv

Les quatre premiers bits du premier octet (1000) indiquent que l'information est une «note off». Les quatre suivants (cccc) indiquent le canal MIDI. Les bits nnnnnnn représentent le numéro de la note à couper.

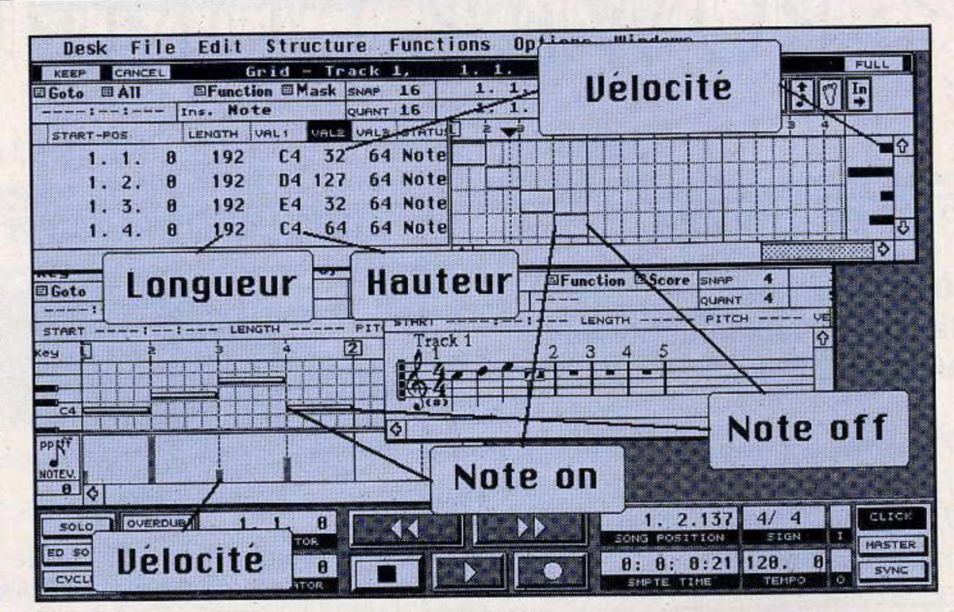
Les bits vvvvvvv ne servent à rien pour ce message qui est généralement nul. Sa présence ne s'explique que par la similitude de structure avec la «note on».

Dans l'exemple, lorsqu'on arrête de jouer le Do 5 en relâchant la touche, le clavier maître envoie: 10000001, 01010100, 00000000, soit 129, 84,0 pour que le générateur sonore cesse de jouer le Do 5. Dans le tableau page 41, vous trouverez les codes «note off» et «note on» avec leur correspondance en décimal et hexadécimal.

Pour bien comprendre qu'une «note on» a besoin d'une «note off», vous pouvez réaliser cette petite expérience. Enfoncez une touche sur votre clavier maître et débranchez le cordon MIDI qui le relie à votre expandeur avant de relâcher la note. Elle va jouer continuellement car elle ne pourra pas recevoir de «note off».

La fréquence d'un son

Le son est un phénomène physique. C'est une vibration mécanique qui se propage dans l'air. La vibration d'une corde, d'une membrane de haut-parleur, ou d'une corde vocale, crée une succession rapide de changement de pression des molécules de l'air. Ces variations de pression successives, appelées aussi ondes, sont alternativement positives puis négatives. La compression et la décompression se communiquent de molécule en molécule, mais l'air lui-même reste bien sur immobile. Dans vos enceintes, l'électro-aimant fait vibrer la membrane du haut parleur qui à son tour fait vibrer l'air et ensuite le tympan de votre oreille. Sur les instruments de musique, les pressions/décompressions se succèdent de façon régulière (contrairement au bruit ou à la voix). Dans ce cas, on dit que l'onde est périodique et l'on nomme période, ou cycle, la succession complète d'une compression/décompression. La fréquence d'une onde sonore mesure le nombre de cycles par seconde. Un cycle par seconde équivaut à un Hertz, unité de mesure de la fréquence. Ce paramètre est primordial en musique car c'est lui qui détermine la hauteur de la note. Chaque fois que la fréquence est multipliée par deux, la hauteur de la note est augmentée d'une octave. Chaque note de musique correspond donc à une fréquence particulière. A titre anecdotique, l'oreille humaine peut en moyenne entendre une fréquence allant de 20 Hertz à 20 KH2. Les musiciens s'accordent souvent sur le La 3, soit 440 Hertz.



Les notes «on» et «off» dans un séquenceur.

Les éditeurs MIDI

La plupart des séquenceurs permettent d'éditer les informations «note on» et «note off» de différentes manières. Mais la position de la «note off» est indiquée en fait sous forme de longueur de note. Dans un éditeur de partition, une distance entre «note on» et «note off» d'un temps joué en 4/4 est représentée sous la forme d'une noire. Dans un éditeur de type grille, l'information est chiffrée. *Cu-Base* de **Steinberg** représente la longueur d'une noire par le chiffre 192. La résolution de ce séquenceur étant de 768 à la ronde, une noire à 4/4 est 768/4=192. Donc pour déplacer une «note off», il faut changer sa longueur. A noter que *Notator/Creator* de **C-Lab** permet la visualisation directe des «notes off». Les éditeurs graphiques de type barre indiquent la «note on» en début de barre et la «note off» à la fin.

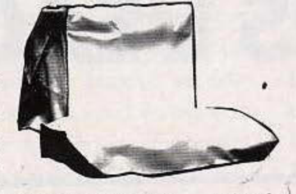
Le mois prochain nous continuerons d'explorer en détail les autres types de «channel voice messages», en attendant si vous avez raté les deux premiers épisodes de cette saga MIDI, n'hésitez pas à les commander auprès du magazine.

Nicolas Jordan ■

Le tableau référencé dans cet article est présenté en page 41.

LA BOUTIQUE A IDEES

HOUSSES DE LUXE



Protégez votre Atari ! En simili cuir blanc, lavables en machine elles vous évitent les pannes dues aux cendres, poussières, ... qui coûtent chères et vous privent de votre micro.

- Housse Clavier 520/1040. Réf.3500 .... 90 F
- Housse moniteur. Réf.3530. .... 125 F
- Housses clavier+moniteur Réf.3510. ....195 F

TAPIS ET HOUSSE SOURIS



Soyez sympa avec votre souris ! Ne la transformez plus en ramasse-poussières avec ce tapis antistatique doublé de mousse.

- *Tapis souris. Réf. 3300 .... 55F*
- *Housse souris. Réf. 3310 .... 85F*
- *Housse+Tapis . Réf. 3320 .... 115F*

KIT DE NETTOYAGE



Votre Atari comme neuf !

Ce kit de nettoyage comprenant : mini-aspirateur et ses diférents embouts, cotons tiges, produits de nettoyage très utiles pour le clavier, l'écran, la souris ou l'imprimante redonnera l'aspect du neuf à votre Atari. La disquette spéciale vous permettra de nettoyer aussi la tête de lecture-écriture de votre lecteur de disquettes.

- *Kit de nettoyage. Réf. 3200.............................. 200F*

101 Jeux d'arcades, de cartes, de réflexion, d'aventures, d'action, de damiers, etc ...



Pour Atari ST, STE, <u>8 mégas tous écrans</u> sur 9 disquettes pleines à craquer ! 8 Méga octets de programmes sélectionnés soigneusement parmi les meilleurs, les plus puissants, les plus originaux et les plus récents d'Europe et U.S.A rigoureusement garantis sans virus. Que vous ayez 1 heure ou une journée devant vous, vous trouverez dans 101 Jeux Atari de quoi vous détendre et vous amuser ! Des heures et des heures de loisirs en perspective.

- *101 Jeux Atari. Réf.3900 .... 495 F*

MOBILIER MICRO INFORMATIQUE



<u>Bureau spécial Micro</u> : Parfaitement ergonomique, il est adapté à tout équipement informatique. Deux tablettes latérales coulissantes (41x29 - 25x29cm) permettent d'y installer une imprimante sur l'une, des boites de disquettes sur l'autre par exemple. Une large étagère (29x72,5cm) sert au rangement. Réalisé en bois, stable et robuste, il bénéficie d'une finition peinture spéciale antireflets et antistatique. Porte clavier avec rebord antifatigue. Dimensions: H 79cm. Plateau : L 77x 41,5 cm. Support clavier : 72,5 X 24cm. Couleur : gris anthracite.

<u>Meuble imprimante</u> : Monté sur roulettes et réalisé avec la même finition et la même qualité que le bureau il est son complément parfait. H 80cm L 60 cm P 40 cm.

- *Bureau spécial Micro. Réf. 3700. .... 1 095 F*
- *Meuble imprimante. Réf. 3710. .... 995 F*
- *Bureau + meuble imprimante. Réf. 3720. .... 1 890 F*

JOYSTICK JET FIGHTER

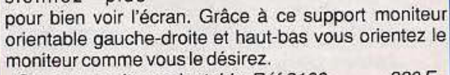
Le joystick pour super scores !

Joystick hyperstable grâce à ses 4 ventouses excentrées, précis grâce à ses 6 microswitches.Deux boutons de tir, un interrupteur de tir en rafale sur la poignée.

Jet Fighter : le coéquipier rêvé pour réaliser des scores fabuleux !

- *Joystick Jet Fighter. Réf. 3620 .... 170 F*

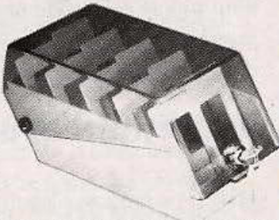
SUPPORT MONITEUR ORIENTABLE



Ayez l'écran bien face à vous ! Ne vous contorsionnez plus pour bien voir l'écran. Grâce à ce support moniteur orientable gauche-droite et haut-bas vous orientez le moniteur comme vous le désirez.

- *Support moniteur orientable. Réf.3100............220 F*

BOITE DE RANGEMENT



N'égarez plus vos disquettes ! Les disquettes sont sensibles à la chaleur, le soleil, les liquides. Mettez-les à l'abri et profitez-en pour les classer. Couvercle teinté, fermeture à clé, cinq intercalaires.

- *Boîte rang. (40 3"1/2).Réf.3420 .... 85F*

BON DE COMMANDE A REMPLIR (OU UNE PHOTOCOPIE) <u>**TRES LISIBLEMENT**</u> ET A ENVOYER ACCOMPAGNE DU REGLEMENT A :

MCM Europe 16 Quai J-B Clément 94140 ALFORTVILLE.

Nom : .... Prénom : ....

Adresse : ....

....

Code postal : .... Ville : ....

<u>Tous nos envois sont en recommandé.</u>

DOM TOM et étranger : 20F de port <u>supplémentaires</u> et paiement par **mandat international uniquement.**

Article	Réf.	Quantité	Prix

<u>Frais de port</u> : 28 F jusqu'à 350 F d'achats, 36 F au dessus, pour mobilier informatique : 120F supplémentaires ....

TOTAL ....

DU NOUVEAU DANS L'EAO
Convivialité avant tout

Avec les nouveaux éducatifs de Micro C et de Lankhor, les vacances scolaires vont pouvoir être mises à profit afin de faire découvrir aux enfants que révision et plaisir peuvent se conjuguer.

Nous l'avions souligné dans le dossier consacré à l'Education assistée par ordinateur (voir Atari Magazine n°31), les éditeurs de logiciels éducatifs doivent offrir aux enfants non seulement des outils pédagogiques, mais aussi des partenaires de jeux et de distraction de qualité.

Micro C et **Lankhor** proposent aujourd'hui six nouveaux éducatifs scolaires tirant pleinement partie des capacités graphiques et sonores de l'Atari. **Micro C** poursuit le développement de sa gamme destinée à la révision du français (CE1 et 4/3). **Lankhor** qui jusqu'à présent s'était cantonnée au seul domaine du logiciel d'éveil, propose la gamme «esprit» pour les classes du primaire.

Afin de mieux vous guider dans le choix des éducatifs, vous trouverez en fin d'article une grille d'évaluation reprenant tous les critères utilisés lors de notre dossier du mois de janvier. Ces critères sélectifs permettent de voir d'un simple coup d'ôil non seulement notre jugement sur les produits cités, mais aussi les divers éléments de notre appréciation.

Apprendre avec Micro C

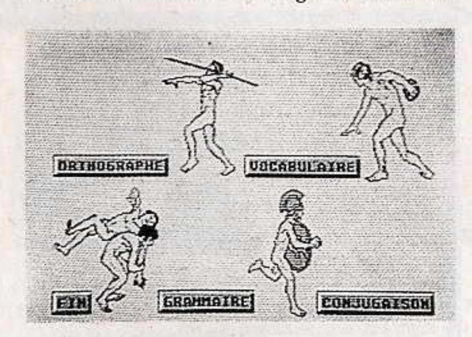
Micro C propose depuis quelques années tout une gamme de logiciels éducatifs destinée aux enfants de tous âges. Malgré l'étendue des thèmes abordés, certains ne sont toujours pas complets. C'est la raison pour laquelle deux nouveaux outils ont pour vocation de faire réviser le long apprentissage du français dispensé en classe de cours élémentaire première année (CE_1) et dans les classes de 4ème et de 3ème. Malgré la «réformette» de l'orthographe, les exercices proposés nous rappellent que la langue française connaît de nombreuses subtilités.



Micro C, français CE[1]: partez gagner la coupe du grand prix de l'orthographe.



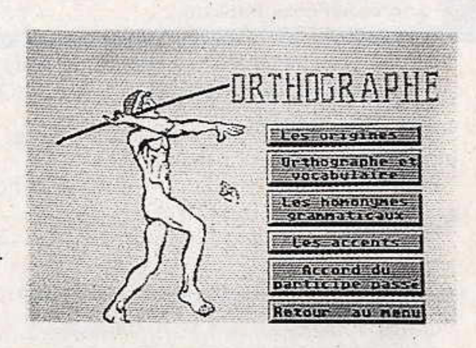
Micro C, français CE[1]: le circuit de la conjugaison permet de parcourir les subtilités de nos verbes.



Micro C, français 4/3: les jeux olympiques illustrent la compétition à mener pour devenir le meilleur en français.

Français CE_1

Le logiciel se compose de deux programmes distincts: le premier permet d'effectuer les exercices, tandis que le second nous donne accès à une grille d'évaluation des progrès de l'enfant. C'est donc après avoir sélectionné le premier que nous accédons à l'environnement de travail du Français niveau CE_1. Apparaît alors un petit garçon assis sur une moto prêt à se lancer sur un des circuit du grand prix: vocabulaire, conjugaison, orthographe, son et lecture. Il est possible de choisir un nouveau championnat ou de revenir à la compétition en cours. La difficulté des exercices variera tout au long de l'année en fonction du niveau atteint. Chaque prix se compose d'un ou plusieurs exercices au terme desquels les points exprimés en kilomètres permettent d'aller quérir la coupe de la victoire. Au début de chaque circuit, une explication est donnée sur l'objectif de l'exercice, sur l'utilisation du clavier et de la souris. Le circuit du vocabulaire propose à l'enfant d'apprendre l'alphabet, de reconnaître les familles de mots et d'identifier les mots contraires comme «gros», «maigre», etc. Le circuit de la conjugaison, quant à lui, permet de se pencher sur les accords entre le sujet et le verbe. Celui de l'orthographe se focalise sur le problème des faux amis: verbe et adverbe. Ainsi l'enfant apprendra à distinguer «à» et «a», «sont» et «son», etc. La course sur le circuit du son est un bon remède contre la dyslexie. Enfin, le circuit de la lecture offre un bon moyen pour comprendre la structu-



Micro C, français 4/3: le sérieux des exercices n'empêche pas une présentation soignée.

re de la phrase. Si à la fin de chaque ensemble de questions les réponses sont correctes, le petit motard en herbe effectue son parcours sans heurt et reçoit les félicitations d'usage. Calqué sur les premières préoccupations de l'apprentissage du français, cet éducatif, qui reste très scolaire, est d'un très bon niveau pédagogique tout en restant attrayant et varié. On notera toutefois le faible niveau de la documentation ainsi que l'orientation du jeu plus destiné aux petits garçons qu'aux petites filles.

Français 4/3

Proposé aux collégiens de classe de 4ème et de 3ème, cet outil est un très bon complément des cours de français à l'aube des toutes premières dissertations. Son objectif répond aux quatre grandes préoccupations majeures de la langue française: l'orthographe, le vocabulaire, la grammaire et la conjugaison. Comme pour rappeler la compétition à mener, le logiciel est illustré par les sports des jeux d'Olympie: le javelot, le disque, la lutte et la course. Malgré cet environnement ludique, le concept de l'ensemble reste très sérieux. Pour l'orthographe et le vocabulaire, trois rubriques sont proposées: des tests pour évaluer le niveau de l'enfant, des rappels de cours riches et synthétiques, des exercices abordés par thème (origine des mots, homonymes, accentuation, sens figuré/ sens propre, etc.) Le module de conjugaison possède une structure équivalente, à la différence près qu'une fonction propose de conjuguer le verbe au temps et au mode de son choix. Pour la grammaire, quatre sujets sont proposés comme préalable: groupe nominal, pronoms, fonctions et phrases complexes. Le logiciel alterne entre des questions à choix multiples où seule est requise la dextérité de l'enfant à manipuler la souris, et des exercices plus élaborés requérant une utilisation plus intensive du clavier. On peut regretter l'absence d'un réel bilan. Cependant, la structure évolutive de chaque partie et l'aide proposée font de ce produit un véritable complément de cours.

L'esprit de Lankhor

L'attrait des logiciels **Lankhor** est d'aller à contre-courant des éditeurs classiques. Au lieu de proposer des outils destinés à la répétition des cours, *L'esprit* est celui de la récréation intelligente. La documentation, bien que succincte, est de bonne qualité et permet une bonne prise ne main des quatre éducatifs testés. Destiné uniquement aux classes du primaire, la gamme *Esprit* n'aborde pour l'heure que le français et les mathématiques. D'autres sujets, comme les sciences naturelles, seront proposés ultérieurement.

L'utilisation du clavier étant exclue, l'ergonomie de chacun des quatre logiciels est excellente. Pour saisir un mot ou des nombres, il faut, à l'aide de la souris, sélectionner les lettres de l'alphabet ou les chiffres représentés en bas de l'écran. Pour savoir si la réponse est bonne, il faut tester cette dernière afin de la valider pour passer à la suite. D'autres points communs entre ces divers logiciels sont à noter: l'utilisation de sons digitalisés de bonne qualité ainsi que la personnalisation du jeu au nom de l'enfant offrent une agréable convivialité.

Lankhor aide les enfants à devenir les égaux d'Einstein ou de Victor Hugo en se distrayant.

Esprits français CE_1/CE_2 et CM_1/CM_2

Les esprits français CE_1/CE_2 et CM_1/CM_2 proposent les mêmes exercices, seul le niveau de difficulté est différent. *Les esprits français* abordent des points précis de vocabulaire, de conjugaison, d'orthographe et de lecture. Prélude de nos aventures, une animation nous montre le prêtre maudit au moment où il ensorcèle le grand livre magique, lequel assure la compréhension de notre langage. Réanimer les six esprits gardiens de notre savoir sera notre objectif.

Pour choisir le thème de l'esprit, c'est-à-dire de l'exercice, il faut tourner les pages du grand livre. Chaque esprit offre un savoir propre: le pluriel, les mots semblables, le verbe oublié, les mots à finir, le texte sacré et la malice.

Les mondes de la chevalerie et de la magie illustrent l'ensemble en toile de fond. Les exercices sont riches et variés. Dans *L'esprit du pluriel*, il nous est demandés de donner le pluriel de mots tels que le ciel, le mal ou encore les coffres-forts, et de replacer des homonymes dans leur contexte (par exemple la cession de biens et la session des examens). Dans *L'esprit du verbe oublié*, il faut conjuguer un verbe au temps demandé, avec un sujet donné.

Durant cet exercice, une table de conjugaison est fournie afin d'aider l'enfant. Dans l'esprit des mots à finir, il faut retrouver des lettres manquantes classés par sons: pour le «f» sont proposés entre autres «ph» comme éléphant ou «ff» comme affiche. *L'esprit du texte sacré* est un texte à trous où l'enfant doit trouver le bon mot.

Placer la première lettre de trois mots dont la fin est identique est le but de *L'esprit des mots semblables*. Enfin, *L'esprit de la malice* nous propose trois petits jeux pour se divertir: *L'alphabet de l'éclair*, *L'infernal mélange* et *Le français musical*.

Le bilan, pour chaque rubrique, se concrétise par le changement de couleur d'un bouclier. La partie est gagnée lorsque tous les boucliers ont changé de couleur. Le sort maléfique sera brisé lorsque toutes vos connaissances se seront améliorées.

Mathématiques CE_1/CE_2 et CM_1/CM_2

Comme pour les esprits français, les deux esprits maths sont identiques, sauf en ce qui concerne le niveau. Une animation soignée nous présente d'entrée de jeu la situation dans laquelle se trouve le grand calculateur d'Euclidia, la planète mère de la galaxie Andromédia. Le calme de ce système solaire a été troublé par les invasions successives des troupes d'Annarkhis la noire. Le grand cal



Esprit français de Lankhor: apprenez les subtilités du français en vous amusant.

culateur a été sabordé. La mission proposée est de libérer les six planètes de la galaxie, en aidant le grand calculateur à retrouver les bons résultats.

Pour accéder aux six groupes d'exercices, il faut sélectionner un des six boutons rouge et vert situés à l'angle gauche du grand ordinateur intergalactique.

Ces six jeux offrent à l'enfant tout une panoplie de divertissements destinée à maîtriser les tables de multiplication, le calcul de l'heure, les conversions d'unités de poids, de longueur et de capacité, le calcul mental et la connaissance des multiples. L'acquisition des réflexes est aussi de la partie.



Esprit maths de Lankhor: aidez le grand calculateur à libérer le système Andromédia en résolvant tous les exercices.

Ces différentes épreuves possèdent des noms aussi évocateurs qu'étranges: la connexion interplanétaire, le labyrinthe multiplicateur, le calculateur du canon laser, la transaction interplanétaire, l'exploration du vaisseau et la console de jeu.

A chaque étape, une petite animation accompagne le bilan du problème posé. La bonne réponse aux exercices fait augmenter les chances de libération d'une des six planètes.

Lorsque les bonnes réponses atteignent 100%, les envahisseurs partent avec femmes et bagages.

Thierry de Rouet ■

Grille d'évaluation des éducatifs

Nom	Français CE_1	Français 4/3	Esprit Français CE_1/CE_2	Esprit Français CM_1/CM_2	Esprit Math CE_1/CE_2	Esprit Math CM_1/CM_2
Editeur	Micro C	Micro C	Lankhor	Lankhor	Lankhor	Lankhor
Catégorie	scolaire	scolaire	ludique	ludique	ludique	ludique
Niveau	CE1	4ème et 3ème	CE_1/CE_2	CM_1/CM_2	CE_1/CE_2	CM_1/CM_2
Monochrome	non	non	non	non	non	non
Couleur	oui	oui	oui	oui	oui	oui
Packadge	très bon	très bon	très bon	très bon	très bon	très bon
Documentation	moyenne	moyenne	bonne	bonne	bonne	bonne
Démarrage	manuel	manuel	manuel	manuel	manuel	manuel
Ergonomie	bonne	très bonne	excellente	excellente	excellente	excellente
Graphisme	assez bon	simple	très bon	très bon	excellent	excellent
Lisibilité	bonne	bonne	bonne	bonne	bonne	bonne
Intérêt	bon	excellent	très bon	très bon	excellent	excellent
Variété	oui	oui	oui	oui	oui	oui
Aide	bonne	excellente	faible	faible	faible	faible
Bilan	très bon	bon	excellent	excellent	excellent	excellent
Notre jugement	favorable	favorable	favorable	favorable	très favorable	très favorable

SYNERGIE & COMMUNICATIONS
Des passionnés du développement

Il fut un temps où développer des applications professionnelles c'était parfois franchement rocambolesque... Aujourd'hui, les pionniers ont mûri. Les développeurs se frottent au marketing.



David René, directeur général de la société Synergie & Communications.

De l'audace, toujours de l'audace... Le premier projet de **Synergie & Communications**, en 1987, représentait déjà 16 mégas de source.
Puis vint l'idée d'aider des auteurs à éditer leurs propres logiciels avec pour seule contrainte de partager leur expérience. Comme le ST ne savait à l'époque que piloter un 68000 à 8 MHz, la société fabriqua sous licence la carte Hypercache à 16 MHz, et n'hésitèrent pas à se lancer dans des projets ambitieux comme des cartes de synthèse vocale. Aujourd'hui, et ce malgré l'essoufflement passager du marché, **S&C** reste fidèle au développement sur Atari. **S&C** c'est aussi un homme, David René, lequel a volontiers accepté de nous présenter les différentes facettes de son activité.

Atari Magazine. La création de votre entreprise répond à quelles motivations?
David René. A l'époque, je trouvais qu'il n'y avait pas assez d'éditeurs sur le marché Atari qui proposaient des produits suffisamment sérieux. Il fallait faire du soft de qualité, d'où le nom **Synergie & Communications:** la synergie des auteurs et la communication qui permet de faire vendre. Donc, avant tout, c'est le manque de softs professionnels, qui entravait la crédibilité de la marque, qui ont été a l'origine de **S&C**.

A.M. Quel a été le premier logiciel professionnel de **Synergie**?
D.R. C'est une vieille histoire, elle remonte à 1987! Il s'agit du Bilan personnel micro informatique, un outil d'aide destiné à la décision du recrutement de cadres supérieurs.
C'est le logiciel le plus cher du marché du monde Atari, à savoir 44 000 F HT. C'est aussi l'un des projets les plus fous, puisqu'il préfigure l'une des premières applications que l'on peut qualifier de multimédia, c'est-à-dire mélangeant sons et images. Même si ce produit ne connaît toujours pas de commercialisation, il reste en 1992 toujours d'actualité, la concurrence n'ayant pas encore réalisé un programme d'une telle ampleur.

A.M. Pour quelle raison le produit n'est-il pas encore sorti?
D.R. Il y a encore environ un an et demi, lors des dernières négociations, s'est posé le problème de son portage sur PC. Au départ tout le monde y croyait. Le PDG de la société, avec qui nous avons développé conjointement le logiciel, a eu tout le loisir de constater en interne son bon fonctionnement ainsi que le respect du cahier des charges. Cependant, des pressions de conseillers ne connaissant que l'univers PC ont empêché la sortie de ce produit sur Atari.

A.M. Une issue est envisagée?
D.R. Non, pas dans l'immédiat. Ici, le problème porte sur la propriété des droits des tests psychologiques qui ne nous appartiennent pas. En fait, nous sommes propriétaires d'une coquille logicielle performante, qui, sans les tests, est une coquille vide. Le produit est aujourd'hui mort-né.

A.M. Quelles sont les sociétés qui étaient prêtes à acquérir ce logiciel?
D.R. BMW, Gervais Danone, la **SNCF**, la **RATP**, etc. En fait, plus d'une centaine de clients potentiels est intéressée par la solution complète comprenant softs et matériels. Certaines de ces sociétés ont eu l'occasion de tester ce produit et de juger la qualité de l'outil fourni aux directions du personnel.

A.M. Pas de version PC en vue?
D.R. C'est ici une histoire de temps. Comme je suis le seul développeur sur ce projet, le portage ne peut être réalisé qu'en un peu plus de six mois de temps. Seize mégas de source ne se modifient pas du jour au lendemain. D'autre part, il est déjà adapté pour fonctionner sur TT.

A.M. Quels sont les outils qui ont permis un tel développement?
D.R. Les images, 500 environ, ont été dessinées à la main sous Degas. C'est un record, à l'époque les scanner n'existaient pas. Les dessins ont été réalisés en trois mois. Avec les outils dont nous disposons actuellement, les dessins pourraient être effectués en huit jours.

A.M. C'est un peu l'histoire de l'évolution du développement?
D.R. Tout à fait. Par exemple au niveau des langages, lorsque j'ai commencé à développer mon premier source, il n'y avait que le *GFA 2.0.* Pour gérer les ressources des boîtes de dialogues c'était la croix et la bannière. Lorsque l'on compare les possibilités actuelles du *GFA 3*, on peut comprendre que les temps de développement se raccourcissent.

A.M. Pour vous, le développement sur les ordinateurs Atari est-il très particuliers?
D.R. Oui, tout à fait. Le défaut majeur au niveau du développement sur Atari porte sur les outils mis à votre disposition. La plupart des développeurs ne sont en réalité que

dès novices, des débutants. Aussi, les programmes qui sont entrepris n'aboutissent pas faute de n'avoir pas respecté des règles élémentaires de programmation. Cependant, leur force est d'avoir réussi à contourner les obstacles, car les documentations ne sont pas légion. Par la suite, ils vont naviguer sur d'autres machines qui possèdent des documentations et des outils de développement suffisamment nombreux pour leur permettre de ne pas rester en panne avec un projet qui, en bout de course, n'est pas commercialisable.

A.M. Vous avez eu des problèmes identiques lors de vos «premiers pas»?
D.R. Oui, mais je suis encore sur Atari car je crois en la force des matériels que les développeurs nous apportent. De plus, je suis convaincu qu'il n'est toujours pas trop tard pour avoir de bons outils de développement.

A.M. Quels ont été les tout premiers logiciels commercialisés par **S&C**?
D.R. Le tout premier n'était pas à proprement parlé un logiciel sérieux, puisqu'il s'agissait de *Rosemary Racoon*, un petit jeu de strip-tease. Son auteur, Bruno Bellamy, est avant tout un dessinateur avant d'être un programmeur. C'est d'ailleurs plus un «bidouilleur», passionné par la programmation. Il a passé beaucoup de temps pour mettre au point son programme, mais il l'a sorti. D'autres auteurs, comme Loic Duval, ont créé eux aussi des programmes de bonne facture. Il s'agit ici de *Maxific 8.1* lequel permet tout un tas de choses: détection de virus, agenda, répertoire, composition de numéros téléphoniques, bloc-notes, conversion d'unités, etc.

A.M. Avez-vous aussi apporté votre contribution?
D.R. Etant biologiste de formation, j'ai réalisé un outil pour consulter la table périodique des divers éléments chimiques. A titre d'anecdote, je peux vous dire qu'un des utilisateurs de ce produit, qui travaille à la centrale nucléaire de Nogent-sur-Seine, me demande régulièrement d'apporter des petites modifications, ce que j'effectue avec le plus grand plaisir.

A.M. Pour vous le ST c'est un nombre restreint d'initiés?
D.R. Non, c'est un produit grand public. Cependant, il souffre d'un manque d'outils professionnels. Par exemple, il n'existe aucune gestion de projets. Au niveau traitement de texte, nous sommes arrivés au top avec le *Rédacteur 3*, avec malgré tout des petites difficultés d'utilisation. Au niveau comptabilité, nous sommes en train de terminer un projet, car les produits actuels souffrent d'un manque de professionnalisme certain. Lorsque l'on compare avec la centaine de progiciels de comptabilité qui existent aujourd'hui sur PC, l'Atari fait pâle figure. Ce qu'il faut avant tout sur Atari c'est un nombre conséquent de logiciels professionnels qui soient finalisés et testés à fond.

A.M. Des programmes souffrent d'un manque de finition?
D.R. Dans la liste des outils professionnels sur Atari, on peut parler du cas de *Calamus SL* qui est sorti deux ou trois mois trop tôt, ce dernier n'étant pas opérationnel lors de sa sortie. De notre côté, ce fut le cas pour *Scigraph 2* qui n'a connu de réelle commercialisation que lorsqu'il fut totalement prêt. Nous avons préféré repousser la sortie de six mois de ce produit plutôt que d'avoir des «plantages». Le but, au fond, c'est de contenter le client et de ne pas porter ombrage à l'image professionnelle des ordinateurs Atari.

A.M. En dehors de *Scigraph*, quels sont les autres produits professionnels que vous proposez?
D.R. Nous avons un système expert, *Expertise*, qui n'est pas tellement connu. Pourtant, ce dernier possède aussi bien un runtime ST qu'un runtime PC. On peut ainsi développer des applications à l'aide d'*Expertise* et les faire tourner indifféremment sur les deux systèmes d'exploitation.

A.M. Antérieurement, vous avez proposé la carte Hypercache.
D.R. Même si Hypercache a été un bon produit, elle n'a pas marché comme nous l'espérions. Un défaut de communication l'a empêchée de décoller au regard du succès de cartes similaires qui sont arrivées sur le marché près d'un an et demi après la notre. Nous l'avons arrêtée, car elle nous a fait perdre de l'argent. Nous nous sommes mouillés dans sa fabrication. En France, c'est très difficile de réussir à produire des cartes hardware en petite quantité. Vous êtes pris dans un étau entre d'un côté les prix, qui pour être intéressants doivent porter sur de grandes quantités, et de l'autre les délais impartis.

A.M. Par ailleurs, vous vous êtes recentrés sur des produits grand public dont certains possèdent un environnement professionnel, comme *BCD Gest*.
D.R. Nous essayons lors de la sortie de nos produits d'apporter une valeur ajoutée, qui est l'environnement développé par **S&C**. Nous sommes plusieurs à nous concerter sur l'aspect convivialité lors de la commercialisation d'un produit. C'est la cas, par exemple, pour notre programme de budget familial comme pour notre progiciel de comptabilité, rebaptis *éS_Compta*.

A.M. Que pensez-vous de l'intuitivité d'une interface?
D.R. C'est purement marketing. Si vous prenez un novice en informatique, votre produit aura beau être intuitif, l'utilisateur aura des difficultés. Lors de la mise au point de notre interface, nous avons essayé de limiter les recherches et les tâtonnements pour arriver à faire fonctionner le programme du premier coup. Ainsi, le manuel d'utilisation n'a en principe aucun intérêt. Cependant, la lecture approfondie du manuel de référence est un passage obligé pour tirer partie des fonctionnalités d'un produit. On peut néanmoins minimiser toutes les manipulations «barbares» pour offrir à l'utilisateur un outil accessible.

A.M. Des projets sont en cours dans vos cartons?
D.R. Oui, mais le problème auquel nous sommes confrontés aujourd'hui est nouveau. Nous sommes encore à l'époque héroïque où l'on peut faire un bon produit en un laps de temps assez court. Cependant, les développements doivent être effectués en équipe, avec des spécialistes pour chaque domaine. Il faut que l'accès aux routines soit libre et non limité à un petit cercle de programmeurs qui gardent jalousement leur invention. Aujourd'hui, nous sommes décidés à donner nos sources aux sociétés prêtes à collaborer avec nous.
L'exemple significatif porte sur un projet de logiciel de dessin qui tarde faute de ne pas avoir librement accès aux différents formats d'images existants.

A.M. Vous pensez pouvoir faire quelque chose?
D.R. Oui, nous pensons depuis longtemps à créer en France une association de programmeurs développant exclusivement sur ordinateurs Atari...
Propos recueillis
par Thierry de Rouet ■

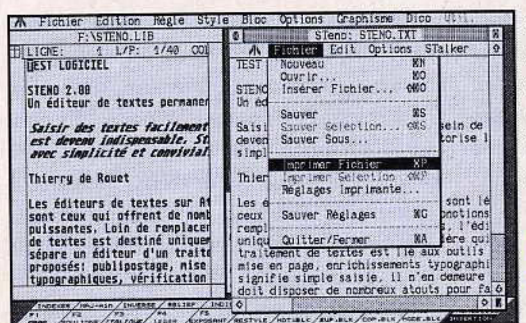
STENO 2.00
Un éditeur de texte permanent

Saisir des textes facilement au sein de toutes vos applications est devenu indispensable. Sténo 2 autorise l'édition de texte avec simplicité et convivialité. Un vrai régal.

Les éditeurs de textes sur Atari sont légion. Cependant, rares sont ceux qui offrent de puissantes fonctions d'édition. Loin de remplacer un traitement de texte, l'éditeur de texte est destiné uniquement à la saisie. La frontière qui sépare un éditeur d'un traitement de texte tient aux outils proposés par ce dernier: publipostage, mise en page, enrichissement typographique, vérification orthographique, etc. L'éditeur de texte est en somme le noyau de base d'un traitement de texte. Mais si édition signifie simple saisie, il n'en demeure pas moins qu'un tel produit doit disposer de nombreux atouts pour faciliter la mise en forme des idées jetées sur le clavier.

Présentation et installation

Sténo 2 est livré avec une documentation claire et didactique d'une trentaine de pages,



Sténo 2 est un éditeur de textes possédant une ergonomie poussée et de puissantes fonctions.

qui permet de maîtriser rapidement le logiciel. L'installation sur disque dur s'effectue manuellement par simple recopie des divers fichiers.

Sténo 2 est utilisable sous deux formes: en application GEM et en accessoire. L'intérêt ici est de se servir de *Sténo 2* en accessoire, un véritable traitement de textes étant préférable lorsque l'on doit traiter des documents volumineux. *Sténo 2* est avant tout un outil d'appoint complémentaire d'autres applications.

Attention piratage!

Certaines sociétés «commercialisant» des logiciels du domaine public proposent depuis plusieurs mois une version 1.0 de Sténo. C'est en contravention avec la loi sur la protection des droits d'auteur que ce produit est aujourd'hui disponible. Malgré la mise en garde de la société Arobace auprès des sociétés en infraction, la première mouture de Sténo est toujours disponible par ce canal de diffusion. Ce type de pratique est nocif pour deux raisons: d'une part, les auteurs ne perçoivent pas le fruit de leur travail et donc, d'autre part, ne proposent pas de nouvelles versions de leurs logiciels aux utilisateurs. Ces derniers sont à leur tour pénalisés. Heureusement, ce n'est pas le cas de Sténo.

Fonctionnement

Fonctionnant entièrement sous GEM, *Sténo 2* se présente sous la forme d'une fenêtre pourvue, sur la ligne d'information, d'un ensemble de menus.

Cette particularité lui permet de s'insérer parfaitement au sein de n'importe quel type d'application respectant les normes de développement GEM. *Sténo 2* présente donc tout l'intérêt d'une application multitâche. Mais il ne s'agit bien évidemment que d'une simulation puisque *Sténo 2* ne peut fonctionner qu'alternativement avec une application.

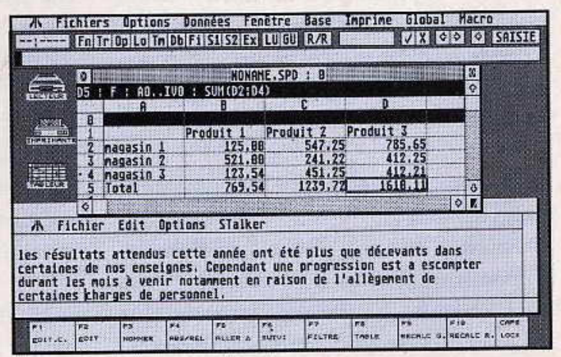
Passerelle vers Stalker

Sténo 2 propose trois niveaux de menus: fichier, édition, option et Stalker.

Stalker (logiciel de communication que nous présentons dans ce numéro) est le complément privilégié de cet éditeur de texte.

Ergonomie de qualité

L'ergonomie de *Steno 2* est irréprochable. Ainsi, la sélection d'un bloc de texte s'effectue par simple déplacement de la souris en maintenant le bouton gauche appuyé. Le fait de presser conjointement sur la touche [Shift] et sur le bouton gauche de la souris permet d'étendre le bloc sélectionné en avant comme en arrière. Pour sélectionner le mot sur lequel le curseur est positionné, il suffit simplement de double-cliquer dessus. Pour effacer un mot ou un groupe de mots, il faut sélectionner la partie du texte à supprimer et saisir le ou les nouveaux mots.



Sténo 2 s'utilise au sein de toutes les applications 100% GEM: ici en action avec Kspread 4, les commentaires des données traitées peuvent être saisis et consultés à loisir.

Un presse papier GEM intelligent?

La fonction couper/copier/coller ne fonctionne sur la gamme STE qu'au sein d'une seule et même application. Aussi, comme dans le cas de notre utilitaire Sténo 2, il est impossible de couper un texte à partir de Sténo 2 et de le coller dans un traitement de texte comme Le Rédacteur 3. Ceci est, à notre avis, une des pistes que les développeurs devraient suivre pour faire évoluer les outils fonctionnant conjointement.

Fonctions complémentaires

Sténo 2 dispose de nombreuses fonctions qui autorisent une mise en forme du document: presse papier (couper, copier, coller), annulation de la dernière tâche effectuée, recherche et remplacement sélectif (prise en compte ou non des majuscules). Comme l'édition de texte implique une gestion par ligne pour respecter le format originel du logiciel (ASCII délimité), toutes les fonctions adaptées à cette caractéristique sont présentes: reformatage paragraphe ou reformatage jusqu'à la fin, aller à la ligne et gestion optionnelle de la césure des mots.

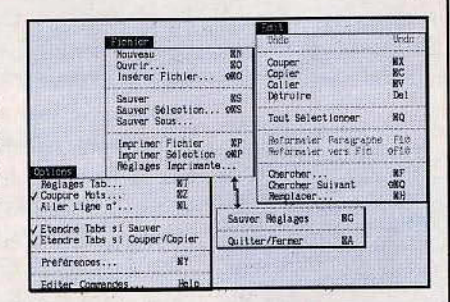
Points forts, points faibles

Sténo 2 permet d'imprimer les textes. La qualité d'impression est sommaire mais la fonction du logiciel n'est pas de mettre en page des documents. Si la francisation est correctement effectuée, la gestion des accents circonflexes fait défaut. Aussi, pour contourner cette limite, vous pourrez, par exemple, utiliser un petit utilitaire malin comme `Accent.PRG` (disponible en téléchargement sur le 3615 ATARI), destiné à corriger ce type de problème. La possibilité de gérer correctement les tabulations est appréciable. Il en est de même pour la modification du buffer, dont la taille par défaut est de 32 Ko, si *Sténo 2* est utilisé comme accessoire.

Un parcours sans faute

Sténo 2 a été mis à l'épreuve durant plusieurs semaines. Le test dont il a fait l'objet n'a révélé la présence d'aucun bogue. C'est un produit fini, utile et performant qui saura vous rendre de grands services.

Thierry de Rouet ■



Les diverses options de Sténo 2, le plus puissant des éditeurs de textes sur Atari.

Sténo 2
Développé par Strata et Gribnif Software
Edité et diffusé par Arobace
Pour Atari ST/STE/MEGA et TT
Ecran couleur et monochrome
Prix: 390 F

De nombreux autres éditeurs de textes vous attendent sur le 3615 ATARI.
Ils sont 100% certifiés domaine public!

STALKER
Un ensemble pour communiquer

De la connexion sur serveurs étrangers à la création de sa propre BBS, en passant par l'automatisation des transferts, voici un logiciel qui devrait ravir les mordus de communication par modem.

S*Talker 3* est un émulateur de terminal très complet, enrichi de possibilités de transfert de fichiers. Il utilise totalement les caractéristiques suivantes: fenêtres, menus déroulants et boîtes de dialogues.

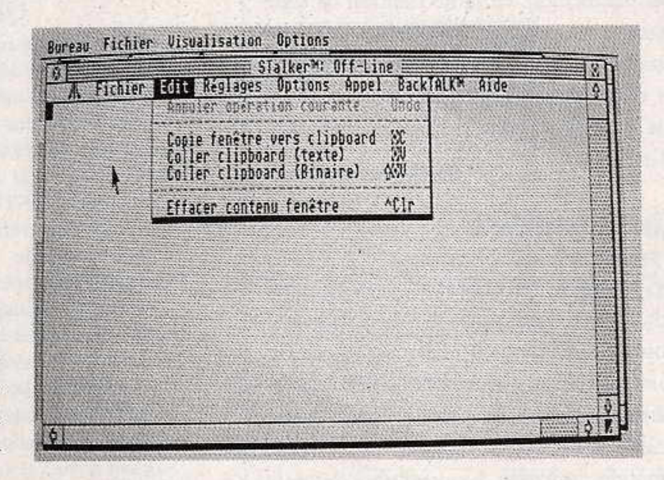
Vue d'ensemble de STalker

Travaillant comme un accessoire de bureau, toutes ses fonctionnalités restent accessibles, même lorsque vous avez lancé d'autres programmes GEM! Bien sûr, vous pouvez aussi lancer *STalker 3* comme un programme normal en le renommant `STalker.PRG`.

Comme principales caractéristiques, on notera un emploi assez particulier des fenêtres GEM, avec des menus intégrés et indépendants en haut de cadre, ce qui permet de travailler avec l'éditeur de texte *STeno 2* (voir l'essai logiciel qui lui est consacré dans ce numéro) sur des fenêtres et des menus indépendants. Par ailleurs, on peut accéder au logiciel *Sténo 2* à l'aide d'une option clavier de *STalker*, et vice versa. Les scripts compilés obtenus avec *BackTalk* (voir plus loin) sont appelés grâce à un menu spécifique de *STalker*.

La mise en œuvre des options de communication est assez aisée si vous avez quelques idées sur la question, mais risque de se révéler ardue pour le néophyte. Heureusement, pour chaque paramètre, la documentation donne des indications claires et précises.

Dans un premier temps, vous déterminerez tous les paramètres de connexion, comme la vitesse en bauds de votre communication, le modem ou la vitesse maximale si vous connectez en direct deux ordinateurs (un Atari avec une station UNIX par exemple).



Transfert du texte vers le clipboard (récupération par STeno) mais aussi vers des fichiers.

Les protocoles reconnus par le logiciel sont bien choisis. On retrouve le VT52 (compatible avec le protocole de l'écran Atari et d'ores et déjà reconnu sur quelques BBS), le VT100 (80 colonnes du minitel 1B, le plus utilisé, car compatible avec tous les serveurs UNIX), ainsi que le PC-ANSI (protocole standard pour une connexion direct sur un PC).

Gestion des protocoles de communication.

En ce qui concerne le téléchargement, *STalker 3* reconnaît les formats ASCII,X,Y et Z Modem, protocoles de transfert que l'on retrouve sur tous les serveurs. Le choix de l'un d'eux dépend principalement de vos réglages et du type de modem dont vous disposez.

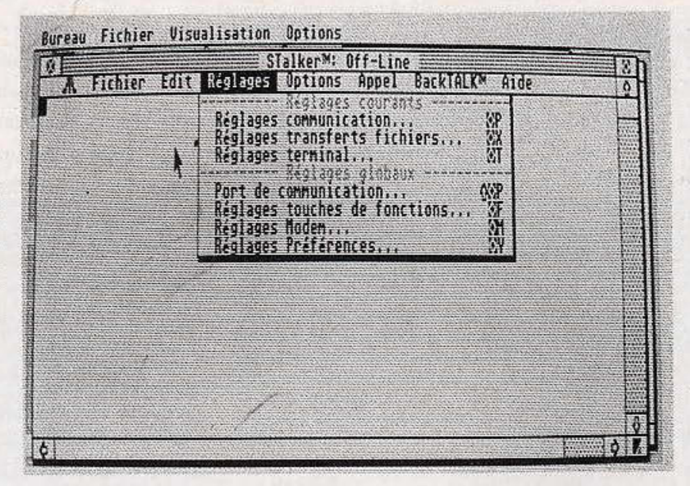
Des transferts en tâche de fond

Les opérations de transferts sont entièrement en tâche de fond, en émission et en réception de données.

Lors des transferts, vous pouvez donc travailler sur votre traitement de texte ou, si vous êtes un fanatique de connexion sur BBS, lire les messages que vous aurez stockés, avant le transfert, dans la mémoire tampon de votre ordinateur. Comme *STalker* est utilisable en accessoire de bureau, vous pouvez bien sûr vous servir d'un programme GEM avec des menus gérant les accessoires!

Faire des économies

Les options «Couper» et «Coller», transfèrent des données entre les applications *Sténo 2* et *STalker*. Sans être très puissantes, elle



L'ensemble des réglages de communication, du modem en passant par le type de serveur et de transfert de fichiers.

rendent de grands services lors d'un usage intensif, par exemple, pour donner une réponse déjà écrite à un correspondant.

L'existence d'un mode «capture» est très pratique, surtout si vous voulez effectuer quelques économies sur vos factures téléphoniques. Tout le texte que vous voyez défiler sur votre écran est automatiquement sauvegardé. Après vous être déconnecté du serveur, vous avez tout le loisir de relire celui-ci.

Les options qui simplifient l'accès aux données et la gestion de l'environnement.

Appel Automatique

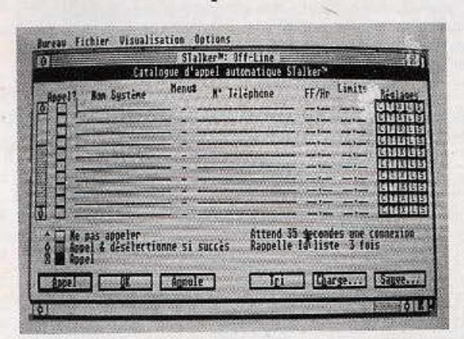
L'appel automatique jusqu'à 30 numéros personnels, dont dix en accès direct par le clavier (Alt 0 à 9), est une petite option appréciable. Elle est idéale pour le zapping entre serveurs, ou pour reprendre une communication interrompue trop brutalement.

Notez que l'on peut générer plusieurs *STalker* en accessoire, en changeant tout simplement le nom du duplicata (par exemple: `STalker2.ACC`). Vous pourrez ainsi vous connecter sur plusieurs serveurs simultanément (seuls les TT ont cette possibilité).

D'autres touches sont programmables. Une vingtaine de fonctions sont ainsi redéfinissables, avec une limitation pour la longueur des textes mais, en revanche, la possibilité d'exécuter ou de chaîner automatiquement des scripts *BackTalk*.

Ecran et téléconférence

Parlons maintenant de l'écran virtuel de 24 à 2000 lignes par 40 à 200 colonnes. Ces chiffres, tirés du manuel, donnent une bonne idée des capacités d'édition. Le nombre de colonnes choisis dépendra des serveurs rencontrés.En revanche, le nombre de lignes offre la possibilité de gérer un mini buffer disponible pour une lecture et une utilisation des textes dans un dialogue multi-utilisateur (par exemple, lors d'une téléconférence où l'on entame une conversation texte à plusieurs). Nous vous conseillons d'essayer au moins une fois!



Appels automatiques: la connexion sur un serveur est prédéfinie et affectée à une touche.

Pour les amateurs de belles mises en pages, l'utilisation des fontes GDOS permet de disposer de plusieurs polices de caractères ayant différentes tailles.

En essayant les fontes «Old English» vous aurez l'impression de lire un livre en langue ancienne.

Cependant, la fonte «Dutch» vous paraîtra sûrement plus lisible.

Qu'est-ce que BackTalk?

Inclus avec *STalker*, *BackTalk* vous permet d'écrire des programmes qui automatisent

Modems et vitesses

Les normes existantes dans le domaine des communications sont nombreuses. Un petit rappel des principales normes retenues par le CCITT (Organisme de normalisation international), vous permettra d'y voir plus clair:

- normes liées au vitesses de communication:
  V22: 1200 Bauds, modem bas de gamme,
  V22Bis: 2400 Bauds, modem bas de gamme,
  V23: 1200/75 Bauds et 75/1200, Minitel,
  V32: 9600 Bauds, vitesse Fax,
  V32Bis: 19200 Bauds,
- normes liées au protocole:
  V42: correction d'erreurs lors de la communication (si le modem serveur et le modem utilisateur en sont effectivement dotés),
  V42Bis: protocole de compression de données sur la ligne (les deux modems doivent en être pourvus).

Il faut surtout retenir que ces numéros, qui fleurissent sur les notices des modems, ne signifient pas que le plus haut chiffre corresponde au meilleur modem.

Cherchez surtout à reconnaître celui qui gère les vitesses et les protocoles dont vous avez besoin (par exemple, pour un fax/modem on retrouve surtout la norme V32, pour l'émulation de Minitel la norme V23, etc.)

Lancement de scripts compilés: on affecte un script à une touche ([Shift], [Alt], 0 à 9).

n'importe quelle opération que vous pourriez effectuer avec *STalker*.

Vous pouvez, par exemple, automatiser des tâches de télécommunication répétitives, comme regarder chaque jour dans votre boîte à lettre électronique, ou encore régler vos transmissions selon les paramètres du système hôte, etc.

BackTalk est un langage de programmation compilé et structuré,
orienté blocs et inspiré du langage C. Bien que la syntaxe en soit différente, les opérateurs et la structure de contrôle sont similaires. Si vous êtes un peu familiarisé avec le C ou le *Pascal*, vous ne devriez avoir aucun problème pour apprendre *BackTalk*. Pour créer des programmes *BackTalk*, vous aurez besoin d'un éditeur de texte capable d'écrire des fichiers ASCII (texte). *STeno*, que nous vous présentons dans ce numéro, est, à cet effet, très pratique.

Le compilateur BackTalk

Le compilateur prend le fichier contenant votre programme *BackTalk* et génère un nouveau fichier dans un format que l'interpréteur de scripts compilés peut comprendre. Ce processus est appelé «compilation», et le fichier créé par le compilateur se nomme «script compilé». S'il y a, dans votre programme, des erreurs de syntaxe qui empêchent la compilation, le compilateur les affiche. Mais il n'y a pas de vérificateur de syntaxe en ligne, l'éditeur étant indépendant du compilateur.

Une fois que vous avez le script compilé, vous pouvez l'exécuter en utilisant la commande «`Lancer script BackTalk...`» dans le menu «BackTALK» de *STalker*.

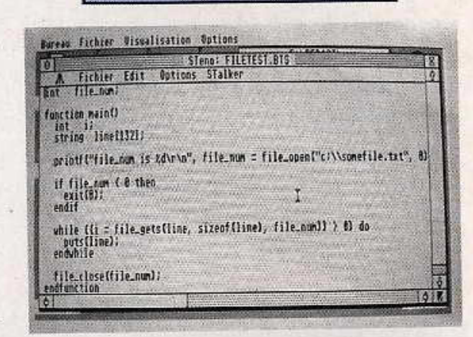
Le compilateur peut être lancé comme un programme indépendant ou comme un accessoire de bureau.

Pour le lancer comme un programme, nommez le fichier `BackTALK.PRG` et double-cliquez dessus. Pour l'utiliser comme accessoire, nommez-le `BackTALK.ACC`, placez-le dans le même répertoire que vos autres accessoires de bureau et relancez votre ordinateur (reset). Le compilateur se trouvera alors dans le menu «Bureau».

Les commandes du logiciel sont contenues dans une bibliothèque qui est appelée au début de chaque script. On remarque d'emblée que l'on peut fabriquer des applications puissantes. Il est possible, par exemple, de gérer un mini BBS personnel. L'option existe bien avec le programme *STalker*, mais les possibilités sont assez réduites et la protection d'accès quasi nulle. Les applications les plus intéressantes sont constituées d'accès automatiques à différents serveurs, durant votre absence, pour aller chercher le courrier ou télécharger des logiciels pendant les heures bon marché.

Hervé Dudognon ■

STalker 3.0 avec BackTalk
Distribués par Arobace
Pour ST/STE/MEGA et TT
Ecran couleur et monochrome
Prix: 690 F

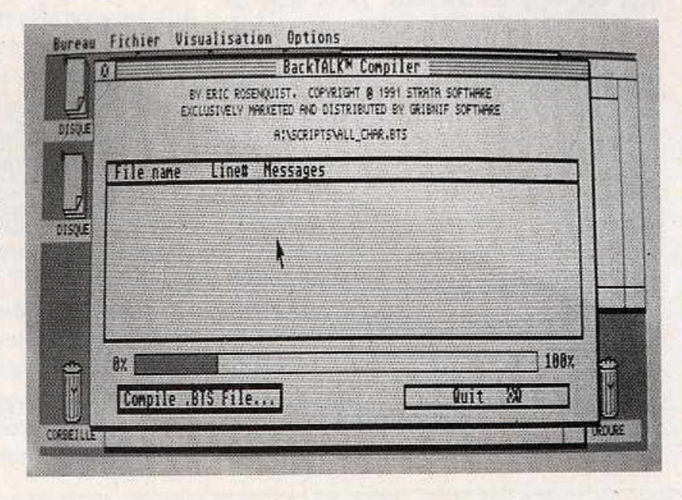


Un script BackTalk sous STeno.

Qu'est-ce qu'un BBS?

Les BBS (Bulletin Board System) ont pour vocation de réunir un groupe de personnes autour d'un ou de plusieurs centres d'intérêts. Ils se présentent sous la forme de serveurs où l'on accède avec un mot de passe ou, pour les nouveaux venus, par l'intermédiaire d'un menu d'information et d'identification assez complet. Le sysop (opérateur système) du serveur validera votre demande d'accès après vérification des renseignements que vous aurez fournis.

Vous pénétrez ensuite dans un monde de passionnés, près à discuter des heures durant (si le sysop est d'accord) sur un sujet précis. Bien entendu, vous n'êtes pas en liaison directe avec tous les inscrits, sauf dans le cas de conférences. Le plus souvent, vous devrez passer par l'intermédiaire d'un système de boîtes aux lettres privées ou ouvertes à l'ensemble des utilisateurs.



Tout ce que l'on peut voir avec le compilateur BackTalk.

MUSIC PACK MIDI

de l'initiation à la création musicale

5.490 F TTC
Prix public conseillé
(hors écran)

Toutes les musiques au bout des doigts !

Grâce au MUSIC PACK MIDI, réveillez le musicien qui sommeille en vous, apprenez la musique en créant vos propres mélodies et vos orchestrations. Conçu pour les musiciens professionnels comme pour les débutants, le MUSIC PACK MIDI réunit 3 produits leaders en un seul package :

• le STE 1040 ATARI, l'ordinateur familial le plus vendu, utilisé par tous les professionnels de la musique (1Mo de RAM extensible à 4Mo, interface MIDI intégrée, 10 interfaces en standard, la plus importante logithèque), l'outil idéal pour s'ouvrir à la micro-informatique musicale ;

• le célèbre clavier BONTEMPI AZ 7500 (clavier MIDI "splitable", 61 grandes touches, 32 sons "PCM Digital", 24 rythmes PCM technologie numérique, 7 voies d'accompagnement, polyphonique (12 notes), 4 effets sonores intégrés) ;

• l'excellent logiciel BIG BOSS Piano (cours de piano classiques et modernes intégrés, clavier guide, défilement de la partition en temps réel, impression des partitions, quantisation).

Avec le MUSIC PACK MIDI, en 10 leçons progressives, assisté en permanence par l'ordinateur, finies les longues heures passées à subir le solfège : apprenez facilement, rapidement et efficacement le piano, l'écriture de la musique.

Avec le MUSIC PACK MIDI, disposez d'un formidable outil de création musicale :
BIG BOSS Piano est également un séquenceur 25 pistes, permettant le chargement simultané de 10 morceaux..

Le MUSIC PACK MIDI est un système évolutif et créatif, doué de toutes les caractéristiques d'un home studio, pour un prix très attractif.

Avec le MUSIC PACK MIDI, de la musique classique au rock le plus sauvage, du rap au reggae, mettez toutes les musiques à portée de vos mains, tous les sons au bout de vos doigts !



En micro-informatique musicale : dites-leur Atari

ATARI FRANCE : 79, avenue Louis-Roche 92238 Gennevilliers Cedex • Tél. Service Informations : 40 85 31 31 ou 3615 code ATARI

LE BLOC-NOTES

De nombreux et bien jolis programmes ont pointé ce mois-ci le bout de la disquette. Tout un tas de petits nouveaux vont déferler avant les mois d'été. Panorama sur le présent et le futur proche.

Lankhor

Ce n'est qu'à la prochaine rentrée que vous pourrez exercer vos dons acérés d'enquêteur hors pair, dans le nouvel épisode des aventures policières du sieur Lange. *Sukya*, le pavillon du thé, vous plongera dans l'atmosphère mystique d'un monastère japonais, sous les cerisiers fleuris d'un jardin zen. Les affreux y seront très méchants et pas du tout «zénés» de ponctuer leur passage en laissant des cadavres un peu partout. En attendant, vous pourrez calmer votre impatience fin mai, avec *Black Sect*, une fantaisie périgourdine déjà sortie sur «huit bits» mais complètement réécrite et redessinée pour Atari. Cette aventure, moins difficile que Maupiti *Island*, vous entraînera sur 34 lieux différents avec une gestion par icône et une évolution de l'image en fonction de l'heure. Et bien sûr de la musique et du son made in **Lankhor.**

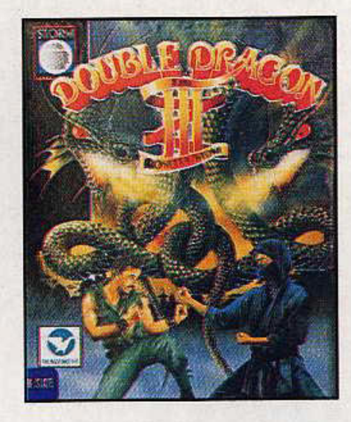
The Shortgrey

Pas vraiment amical le «petit gris» que vous incarnez. Pas vraiment l'apparence d'un escargot non plus, car il possède une machine, le shortwatt, et même un pistolet, le skousik, qui lui permet de revêtir l'apparence humaine. Tout frais débarqué de votre soucoupe, vous chercherez donc à parcourir le vaste monde (environ 300 écrans) créé par l'équipe d'**Accrosoft**, dont le développement (entièrement en français) a duré près d'un an.

Façon *Sierra* ou *Lucas*, vous baladez votre personnage en pointant une direction à l'écran. Une dizaine de verbes usuels présents à l'écran permettent d'agir sur l'environnement. Des scènes explicatives, non interactives, s'intercalent de temps à autre pour faire avancer l'intrigue. Le plus remarquable dans cette préversion, est incontestablement la qualité des graphismes, avec des premiers plans ombrés, de jolis contrastes et une foule de petits détails colorés. Le jeu se résout en cinq ou six heures quand on possède la solution, le temps de devenir complètement accro de ce «petit gris» là!

Double Dragon III

Blaf! ping! paf! Voici les frères Lee aux poings d'acier dans le fils de la revanche du numéro 2 de la plus célèbre castagne de tous les micros. Il vont chercher



l'insaisissable «Rosette Lapier re», comme dit la notice, mais tout cela n'est qu'un prétexte pour bousiller du vilain et endommager son joystick.

Editeur: Storm
Distribué par Ubi Soft
Pour Atari ST/STE
Ecran couleur
Prix: 259 F
Genre: castagne
Avis: cals assurés et phalanges douloureuses pour cet agréable karaté-game, jouable à deux.

Heimdall

Il y a bien des siècles, Odin le dieu roi de Valhalla, créa le peuple Viking. Loki, le dieu du mal, déroba l'épée d'Odin, la lance de Frey et le marteau de Thor pour affaiblir ses adversaires, et les cacha sur terre. Un seul viking allait pouvoir aider les dieux à récupérer leurs attributs essentiels dispersés: le farouche et astucieux Heimdall.

La première partie se présente comme un jeu d'adresse avec le lancer de hache, la course au sanglier et le combat en bateau. Ces épreuves, si vous les passez avec succès, vous permettront d'avoir un choix plus vaste pour former votre équipe d'aventuriers.

Dans la seconde partie, le corps du jeu, vous pourrez emmener trois personnages sur chaque île que vous aborderez. Les scènes que vous découvrirez alors sont en 3D isométriques avec un saisissant effet de perspective et un scrolling doux et fluide de toute beauté. Les décors, les personnages sont fouillés, nets et très colorés, tout à fait dans le style des dessins animés de Don Bluth (le graphiste Jerr O'Carrol a d'ailleurs travaillé avec celui-ci). En chemin, vous résoudrez tout un tas d'énigmes avant de dénicher les outils divins, vous vous équiperez dans les auberges, vous découvrirez de nombreux objets, des armes, des trésors, des potions, des parchemins, bref, tout ce qui fait le pe-



tit train-train ordinaire d'un aventurier d' «heroic-fantasy». Sans oublier les méchants qui font rien qu'à essayer de vous taper sur la tête quand vous avez oublié votre casque... Les combats se font en temps réel avec jet de sort éventuel. Votre adversaire est vu de face, prêt à foncer et, un peu comme dans *Elvira*, il vous faut décider du moment précis de l'attaque ou de la para-

de. Même si la gestion des cinq disquettes est de temps à autre un tantinet agaçante, cet excellent jeu vaut à la fois pour sa profondeur et la beauté de ses dessins.

Editeur: Core design
Distribué par Ubi soft
Pour Atari ST/STE
Ecran couleur
Prix: 299 F
Genre: jeu de rôle
Avis: réalisé avec soin, doté de superbes graphismes proches du dessin animé, Heimdall est un jeu de rôle, vif et coloré, plein de joyeux combats et d'énigmes.



Psyborg

Ce jeu reprend le principe de *Trailblazer* qui proposait de guider une boule sur un chemin composé de dalles rectangulaires possédant des propriétés spéciales comme une accélération soudaine, un saut dans l'espace ou bien un effet de glue. On ne peut qu'admirer la rapidité de l'affichage et la finition du produit. Cependant, malgré les différentes options (bonus, temps, magasins, etc.), le jeu reste quelque peu monotone.

Edité et distribué par Loriciel
Pour Atari ST/STE
Ecran couleur
Prix: 269 F
Genre: jeu de réflexes
Avis: bien réalisé, cette course de vitesse, parsemée d'embûches et de bonus, reste un peu trop abstraite et répétitive.



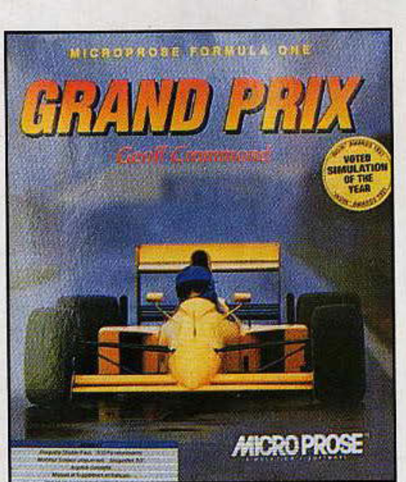
En vrac

La firme anglaise, **Revolution**, risque d'en provoquer une bien belle avec *Lure of the temptress* (le piège de la tentatrice), une grande aventure interactive digne des meilleurs jeux de chez **Lucas ou Sierra**. Les scènes animées, et fort joliment colorées, possèdent des graphismes fins et soignés, proches de ceux de chez **Delphine**, avec de jolis angles et de belles plongées. Le système de jeu est assez original: les personnages ont une vie propre et se baladent dans le jeu. Dès le début, tandis que vous effectuez certaines actions, vous pouvez commander à votre aide de camp, le fidèle Ratpouch, d'effectuer certaines actions. Tous les détails prochainement.

Microprose qui vient de sortir *Knights of the sky*, ajoute au ST deux excellents titres: *Grand prix formula one*, une course de bolides incluant la gestion de l'écurie et de la saison automobile, tout en graphismes calculés ultra rapides, et *Special forces* dans lequel vous prendrez la tête d'un commando de quatre hommes pour des missions secrètes en territoire ennemi. A la fois stratégique et bourré d'actions, un peu dans la lignée de Tank *Platoon*, *Special forces* se présente comme un joli wargame avec des graphismes très fins. Les fans de *Powermonger* (distribué par **Electronic Arts**) vont se régaler avec l'extension, *World war one*. Cela leur permettra d'améliorer leurs performances en élevant le conflit à l'échelle mondiale.

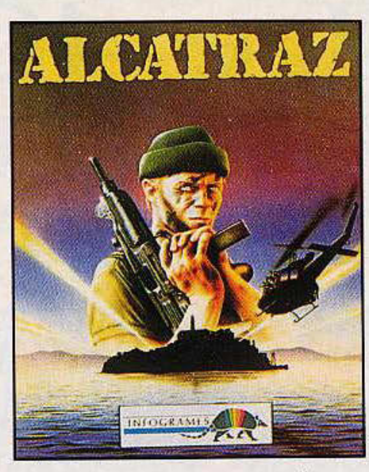
Sous le label **Ubi soft**, **Hudson** a adapté *Bomber man* un programme tournant sur consoles. Ce n'est pas tant le jeu en solo, vif, amusant, plein de logique et tout à fait réussi, qui retient l'attention, mais plutôt le fait que ce jeu sera vendu avec un adaptateur de joystick permettant de connecter jusqu'à cinq joueurs. Celui-ci servira également de protection au jeu. Pour avoir testé la version micro (qui portera le nom de *Dyna Blaster* dans les locaux d'**Ubi**, nous pouvons vous assurer que c'est tout à fait hilarant et très bien conçu, de quoi rabibocher les familles désunies autour d'un écran.

Chez **Ocean**, quatre grands titres sont prévus pour le mois de mai: *Hook*, un jeu d'aventure dans lequel vous incarnez Peter Pan; *Adams Family*, un gigantesque jeu de plate-forme; *Push over*, un puzzle/game avec 250 tableaux, et le déjà fabuleux, le très attendu et mythique *Epic*.



Alcatraz

Miguel Tardiez, le chef du cartel de la drogue, s'est emparé de l'île d'Alcatraz. Vous êtes l'un des deux membres du commando chargés d'éliminer ce redoutable narco-trafiquant.



Votre mission comporte plusieurs phases. Vous devrez franchir le chemin de ronde; investir le bâtiment des gardes; déposer deux bombes à retardement dans la bâtiment industriel; escalader les murs de la prison; neutraliser les gardes du corps de Tardiez et, enfin, accompagné de l'ignoble individu, rejoindre l'hélicoptère. La première phase extérieure se joue sur deux demi écrans coupés horizontalement. Si vous jouez seul, vous alternez vos personnages tandis qu'à deux joueurs, ce qui est nettement plus stressant, vos actions sont simultanées. Tout une panoplie d'armes est à votre disposition (lance-flammes, couteau, fusil, grenades, etc.), et il faudra choisir les plus appropriées aux situations chaudes que vous allez rencontrer. Une fois que vous aurez investi un bâtiment, la vue changera et présentera cette fois un écran coupé en deux verticalement, montrant des couloirs et des pièces dans laquelle un nettoyage par balles s'impose.

Léopold Braunstein

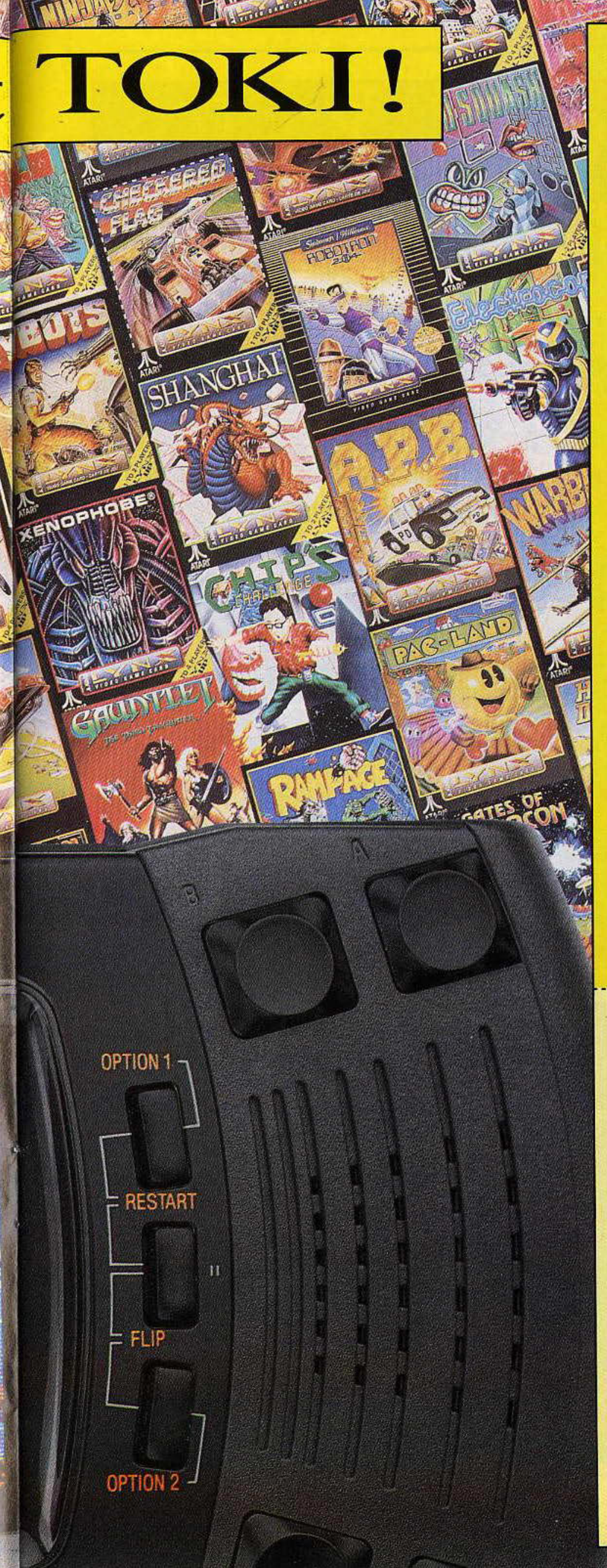
Edité et distribué par Infogrammes
Pour Atari ST/STE
Ecran couleur
Prix: 259 F
Genre: jeu d'action
Avis: le jeu à deux est très plaisant et rapide. Alcatraz possède une bonne ambiance glauque servie par de bons graphismes et une action variée.

Le 40e jeu Lynx, c'est

Attention les yeux...



TOKI !



Avec la console ATARI LYNX, le jeu vidéo a fait un bond prodigieux! La nouvelle console LYNX 16 bits révolutionne le monde du jeu vidéo. Portable et tout en couleur, le LYNX est un remarquable bijou technologique. Tous les grands hits des jeux d'arcade et des nouveautés époustouflantes sont au rendez-vous : déjà 40 jeux passionnants au catalogue. En couleur et avec un son stéréo qui chahute. Le dernier-né, c'est le célèbre Toki, l'homme singe.

40 SUPER JEUX DISPONIBLES

APB	AWESOME GOLF*
BILL AND TED'S*	BLOCK OUT
BLUE LIGHTNING	CALIFORNIA GAMES (4)
CHECKERED FLAG	CHIP'S CHALLENGE
CRYSTAL MINES II*	ELECTROCOP
GATES OF ZENDOCON	GAUNTLET
HARD DRIVIN*	RAMPAGE
ISHIDO*	KLAX
MS. PAC MAN	NINJA GAIDEN
PAC LAND	PAPER BOY
ROAD BLASTERS	ROBO-SQUASH
ROBOTRON 2084*	RYGAR
SCRAPYARD DOG	SHANGAI
SLIME WORLD	STUN RUNNER
TOURNAMENT CYBERBALL*	TURBO SUB
TOKI*	CHESS CHALLENGE
VICKING CHILD	WARBIRDS
XENOPHOBE	XYBOTS*
ZARLOR MERCENARY	jeux à partir de 250 F TTC

*Dernière sortie

BIENTOT SUR VOS ECRANS

BASEBALL HEROES	BASKET BRAWL
CABAL	CASINO
DIRTY LARRY RENEGADE	DRACULA
GEO DUEL	HOCKEY
HYDRA	HYPERDROME
LEMMINGS	NFL FOOTBALL
PINBALL JAM	PIT FIGHTER
RAIDEN	SHADOW OF THE BEAST
SPACE WAR	STRIDER
SUPER SKWEEK	VINDICATORS

DES ACCESSOIRES POUR BICHONNER VOTRE LYNX

Câble Comlynx, pochette, attaché-case, adaptateur de voiture, pare-soleil, pack batteries.

Prix généralement constatés au 01/04/92.

DECOUVREZ LES SECRETS DU LYNX ET DE SES JEUX

Lisez le numéro spécial LYNX d'Atari Magazine. Tous les jeux disponibles et ceux à venir sont passés en revue. Un ouvrage bourré de trucs et astuces. Du jamais vu, de l'inédit! **30 FF** seulement (frais d'envoi compris). Vendu uniquement par correspondance.

Adresser un chèque de 30 FF accompagné du bon de commande dûment renseigné à : ARTIPRESSE/ATARI MAGAZINE - 79, avenue Louis Roche 92238 Gennevilliers CEDEX

Nom

Prénom

Adresse

Code postal

Ville

ATARI

ATARI FRANCE - 79, avenue Louis Roche - 92230 Gennevilliers
Tel.: (1)40.85.31.31 - 3615 ATARI

L'ARCADE A SON TOP
Toki et SuperSkweek sont arrivés

Il manquait au Lynx quelques personnages clés du jeu vidéo, quelques stars incontournables, capables de damner le pion à des Sonic et autres Mario. En voilà deux d'un coup: Toki et Skweek!

Toki

A l'heure où vous lirez ces lignes, l'événement *Toki* sera chez votre revendeur de jeux Lynx. Le moins que l'on puisse dire c'est que les aventures de cette homme-singe étaient très attendues des Lynx maniaques. A force d'espérer, on finissait par s'inquiéter. *Toki* allait-il décevoir? Quelle que soit votre impatience envers *Toki,* vous ne serez pas déçu. *Toki* sur Lynx est tout simplement grandiose, la meilleure adaptation à ce jour tous micros et consoles confondus.
Le diabolique Voodimeklo, non content d'avoir kidnappé la princesse Miho, vous a lancé un sort fâcheux en vous transformant en singe, espérant ainsi vous décourager de toute tentative pour rattraper votre princesse. Erreur fondamentale de sa part puisque vous avez maintenant deux bonnes raisons de le poursuivre.
Car pour retrouver votre apparence humaine, il vous faudra inévitablement tuer Voodimeklo et sauver la princesse.



Toki: le premier des maîtres gardes.

Toki est un fantastique jeu d'action à scrollings multidirectionnels, un peu façon *Viking Child.*
Pour rejoindre la forteresse de Voodimeklo, il faudra traverser des mondes aussi dangereux que diversifiés, peuplés d'animaux étranges et d'humeur chagrine ou des douves remplies de piranhas et autres créatures aquatiques carnivores. De plus, à la fin de chaque niveau, vous affronterez l'un des maîtres gardes de Voodimeklo avant, bien sûr, d'affronter au tableau final, Voodimeklo lui-même.
Tous les ingrédients d'un bon jeu d'action sont présents: variété des niveaux et des créatures, monstres de fin de niveau, tirs multiples, bonus et objets divers à ramasser.

Toki: un second niveau très aquatique.



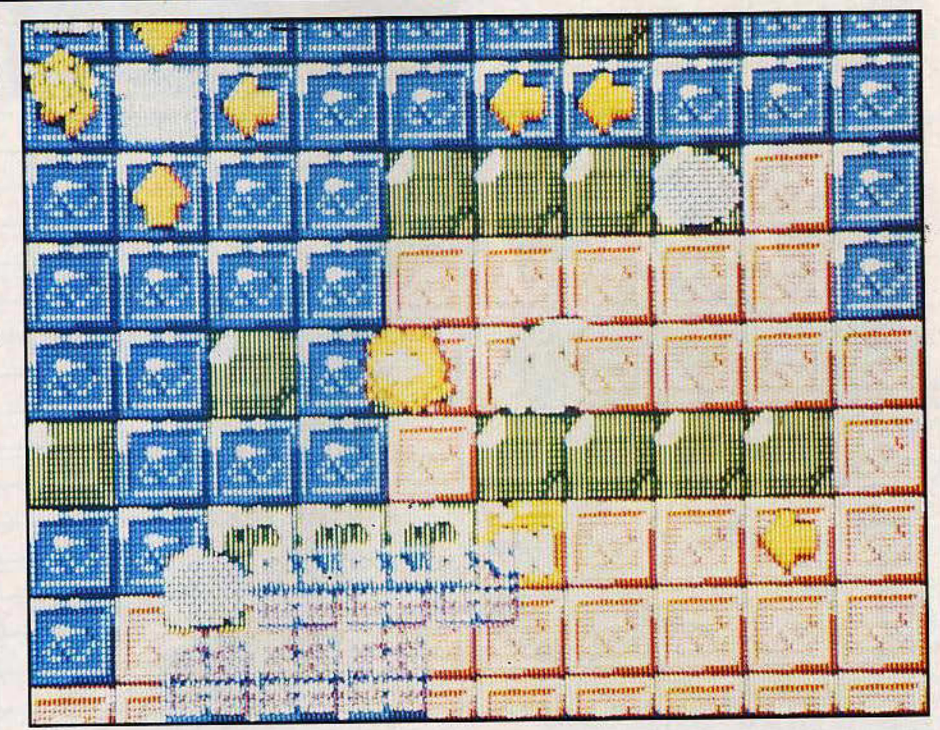
Toki: des sprites gigantesques.

dépense en vaut très largement la chandelle. *Toki* est le meilleur jeu d'action sur Lynx...

Toki
Edité par Fabtek/Atari Corp.
Distribué par Atari France
Prix 290 F
Note Globale: 100%

SuperSkweek

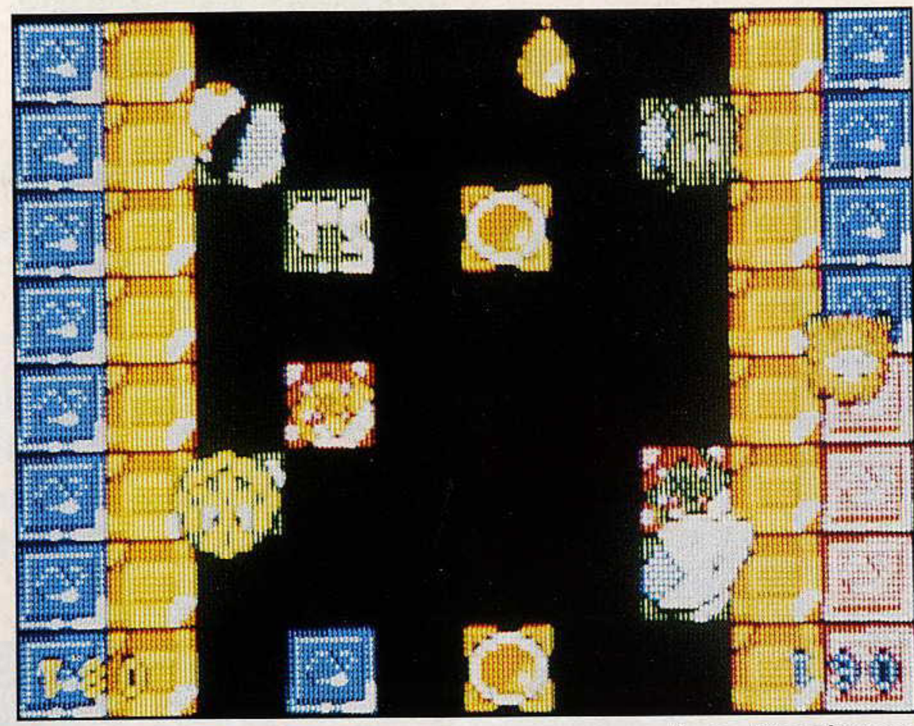
SuperSkweek est un titre que nous attendions avec impatience pour deux raisons. D'abord ce jeu est présent sur toutes les consoles et micros du marché et risque donc de servir de point de comparaison aux acheteurs hésitants (mais comment est-ce possible?) entre un Lynx et une autre console.



SuperSKweek: attention aux fantômes.

La réalisation est à la hauteur de toutes nos espérances: scrollings rapides et fluides, nombreux sprites superbement dessinés et animés, graphismes soignés et colorés, excellente maniabilité du personnage, des niveaux d'une difficulté progressive (du coup le jeu convient aussi bien aux maîtres de l'arcade qu'aux débutants), sons digitalisés. *Toki* est une remarquable démonstration des fantastiques capacités du Lynx et marque un nouveau standard de qualité pour les jeux Lynx.
La variété des niveaux et leurs difficultés parfaitement dosées assurent à ce jeu une longue durée de vie, ce qui est très rare dans cette catégorie.
Courrez immédiatement chez votre revendeur, vous ne le regretterez pas. La

Ensuite, *SuperSkweek* est la première réalisation française sur Lynx, puisqu'il a été programmé par l'équipe de **Loriciel**.
Avant d'aborder l'aspect réalisation, rappelons les principes de ce jeu d'arcade qui tient de *Pac Man*, *Bombuzal*, etc.
Le but est de conquérir cinq îles en repeignant en rose les cases composant les cinquante univers de chaque île. Pour éviter la monotonie, certains niveaux exigent de Skweek qu'il remplisse une autre mission comme, par exemple, celle de libérer les Skweezettes (femelles Skweek) retenues prisonnières.
Evidemment, chaque univers est gardé par des monstres aussi farfelus que dangereux. Certains univers disposent de boutiques permettant à Skweek de s'acheter quelques armes redoutables qui lui faciliteront la tâche.
La réalisation est excellente et les programmeurs se sont amusés à expérimenter toutes les capacités du Lynx. On a donc droit à de spectaculaires effets de zoom ou de scrolling, des écrans tirant parti des 4096 couleurs du Lynx, etc. Les graphismes sont fins et bourrés d'humour. La maniabilité de Skweek est bonne. Incontestablement la version Lynx de *Skweek* est de loin la plus réussie et la plus riche en effets visuels de toutes sortes. Ajoutez à cela 250 tableaux différents, un système de codes permettant de reprendre une partie là où on l'a abandonnée, un mode deux joueurs, un mode où les tableaux se succèdent aléatoirement, des petites séquences animées distrayantes entre chaque tableau et vous obtenez, au final, un jeu d'arcade tout à fait irrésistible...



Super SKweek et ses graphismes très sympas.

SuperSKweek: des monstres et des bonus.

Super SKweek: des petits dessins animés entre les niveaux.





SuperSkweek
Edité par: Loriciel/Atari Corp.
Distribué par Atari France
Prix: 290 F
Note Globale: 89%

Tableau de l'article MAO

Correspondance de "Note on" et "Note off"

Binaire	Décimal	Hexadécimal	MIDI
10000000	128	80	Note off canal 1
10000001	129	81	Note off canal 2
10000010	130	82	Note off canal 3
10000011	131	83	Note off canal 4
10000100	132	84	Note off canal 5
10000101	133	85	Note off canal 6
10000110	134	86	Note off canal 7
10000111	135	87	Note off canal 8
10001000	136	88	Note off canal 9
10001001	137	89	Note off canal 10
10001010	138	8A	Note off canal 11
10001011	139	8B	Note off canal 12
10001100	140	8C	Note off canal 13
10001101	141	8D	Note off canal 14
10001110	142	8E	Note off canal 15
10001111	143	8F	Note off canal 16
10010000	144	90	Note on canal 1
10010001	145	91	Note on canal 2
10010010	146	92	Note on canal 3
10010011	147	93	Note on canal 4
10010100	148	94	Note on canal 5
10010101	149	95	Note on canal 6
10010110	150	96	Note on canal 7
10010111	151	97	Note on canal 8
10011000	152	98	Note on canal 9
10011001	153	99	Note on canal 10
10011010	154	9A	Note on canal 11
10011011	155	9B	Note on canal 12
10011100	156	9C	Note on canal 13
10011101	157	9D	Note on canal 14
10011110	158	9E	Note on canal 15
10011111	159	9F	Note on canal 16

Disquette spéciale listings

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer les programmes qui vous sont présentés dans Atari Magazine n°34. Cependant, vous pouvez vous les procurer soit en les téléchargeant sur le 3615 ATARI, soit en commandant la disquette spéciale listings grâce au bon de commande encarté à la fin de ce magazine.
Cette disquette comprend, en plus des listings, les quatre versions (GFA, C, STOS, Assembleur) du jeu de puzzle spécialement développé à l'occasion de notre dossier langage.
Alors n'hésitez pas...

PROGRAMMEZ EN C
Gestion du scrolling vertical hard du STE

Les capacités graphiques du STE permettent de réaliser des scrollings verticaux absolument parfais. Essayez, vous ne serez pas déçu.

Un peu de théorie

La mémoire écran ST occupe 32000 octets. Son adresse est variable. Le système d'exploitation du STE détermine l'emplacement de la mémoire vidéo en fonction de la mémoire disponible. La fonction système `Physbase()` permet de connaître l'adresse de la zone écran. La fonction `Setscreen()` permet de modifier l'adresse de la mémoire écran.

```
Setscreen(adr_log, adr_phys, res)
long adr_log;
long adr_phys;
int  res;
```

Le paramètre `adr_log` modifie l'adresse de la mémoire écran logique, c'est-à-dire l'adresse vidéo qui va être prise comme référence par les fonctions graphiques. Le paramètre `adr_phys` modifie l'adresse de la mémoire écran physique, c'est-à-dire l'adresse de la page vidéo qui va être affichée sur l'écran. En utilisant des adresses différentes pour l'écran logique et l'écran physique, il est possible d'afficher une image tout en effectuant des opérations graphiques dans une autre image. Le paramètre `res` permet de changer la résolution graphique (0=basse résolution; 1=moyenne résolution; 2=haute résolution). Il est possible de changer la résolution dans un programme TOS, mais pas dans un programme GEM. La fonction `Setscreen()` ne modifie pas les variables système de GEM, qui continue à travailler dans l'ancien mode graphique. Si un paramètre ne doit pas être modifié, il faut lui donner la valeur -1.

```
Setscreen(adr_log,adr_phys,-1);
```

Afin d'éviter un effet graphique très désagréable, le gestionnaire vidéo ne change l'adresse de la mémoire écran qu'après avoir dessiné l'image écran courante. Pour éviter une bavure vidéo, il faut donc attendre que le changement d'adresse s'effectue. Cela se fait avec la fonction `Vsync()`.

```
Setscreen(adr_log,adr_phys,-1);
Vsync();
```

Le gestionnaire écran du STF peut positionner l'adresse de l'écran à 256 octets près. Cela peut servir pour divers applications, mais pas pour réaliser un scrolling vertical. Le gestionnaire écran du STE peut positionner l'adresse de l'écran à l'octet près. Cela ouvre la voie à de nombreuses utilisations, dont la réalisation d'un scrolling vertical pixel par pixel.

Réalisation d'un scrolling vertical

La mémoire écran haute résolution occupe 32000 octets. Elle est organisée en 400 lignes de 80 octets. Pour réaliser un scrolling vertical, il faut créer une mémoire écran virtuel de taille supérieure à l'écran physique. Le scrolling se fait en positionnant l'adresse de l'écran physique sur le début de l'écran virtuel, puis en incrémentant cette adresse de la longueur d'une ligne (80 octets en haute résolution).

```
                              ECRAN VIRTUEL
                          +-----------------+
                          | AAAAAAAAAAAAAAA |
Adresse de                | BBBBBBBBBBBBBBB |
l'écran physique ===>     | CCCCCCCCCCCCCCC |
                          | DDDDDDDDDDDDDDD |
IMAGE AFFICHEE SUR        | EEEEEEEEEEEEEEE |
L'ECRAN PHYSIQUE          | FFFFFFFFFFFFFFF |
+-----------------+       |                 |
| GGGGGGGGGGGGGGG |
| CCCCCCCCCCCCCCC |
| HHHHHHHHHHHHHHH |       |                 |
| DDDDDDDDDDDDDDD |       | IIIIIIIIIIIIIII |
| EEEEEEEEEEEEEEE |       | JJJJJJJJJJJJJJJ |
| FFFFFFFFFFFFFFF |       | KKKKKKKKKKKKKKK |
| GGGGGGGGGGGGGGG |       | LLLLLLLLLLLLLLL |
+-----------------+       +-----------------+
```

La fonction `malloc()` permet de réserver une zone mémoire de taille quelconque. Une zone mémoire de 64000 octets correspond à un écran virtuel d'une taille de 640x800 pixels. Son adresse est stockée dans la variable `adr_tampon`: `adr_tampon=malloc(64000);`

La fonction `set_ligne()` permet de positionner l'adresse de la mémoire écran à une ligne quelconque de l'écran virtuel. Elle calcule l'adresse de la ligne en multipliant le numéro de la ligne par le nombre d'octets par ligne (80 en haute résolution), et en ajoutant le résultat à l'adresse de l'écran virtuel. La fonction `Vsync()` bloque l'exécution du programme tant que le gestionnaire d'écran ne modifie pas l'adresse de l'écran physique.

```
void set_ligne(l)
int l;
{
  long adr_ecr;

  adr_ecr=adr_tampon+l*80;
  Setscreen(adr_ecr,adr_ecr,-1);
  Vsync();
}
```

Scrolling vertical pixel par pixel

La routine scroll1() effectue un scrolling vertical pixel par pixel de l'écran virtuel. L'image affichée sur l'écran est effectuée des lignes 0 à 399 de l'écran virtuel, puis des lignes 1 à 400, 2 à 401, 3 à 402, .. jusqu'aux lignes 400 à 799.

```
void scroll1()
{
 int ligne;

 for (ligne=0; ligne<=400; ligne++)
   set_ligne(ligne);
}
```

La routine scroll2() effectue un scrolling vertical pixel par pixel dans le sens inverse de scroll1(). L'image affichée est constituée des lignes 400 à 799, puis 399 à 798, puis 398 à 797, pour finir aux lignes 0 et 399.

```
void scroll2()
{
 int ligne;

 for (ligne=400; ligne>=0; ligne++)
   set_ligne(ligne);
}
```

Restauration de l'adresse de l'écran physique

Si un programme modifie l'adresse de la mémoire écran, il doit impérativement la remettre à sa valeur initiale avant de retourner à GEM. Sans cela, il risque d'y avoir un effet graphique surprenant et un superbe plantage. L'adresse initiale de la mémoire écran peut être déterminée avec la fonction Physbase(). La fonction main() stocke l'adresse de l'écran dans la variable old_ecran. Elle utilise la fonction Setscreen() pour restaurer l'adresse de la mémoire écran.

```
void main()
{
 long old_ecran;

 /*  STOCKAGE MEMOIRE ECRAN  */
 old_ecran=Physbase();
 /*  APPEL ROUTINE DE SCROLLING  */
 scroll();
 /*  RESTAURATION MEMOIRE ECRAN  */
 Setscreen(old_ecran,old_ecran,-1);
}
```

Augmentation de la vitesse de scrolling

La vitesse de scrolling peut être considérablement accélérée en augmentant le nombre de lignes incrémenté à chaque étape du scrolling. Un scrolling pixel par pixel d'une image haute résolution se fait en 400 étapes. Avec 200 étapes de scrolling et une incrémentation de 2 pixels, la vitesse de scrolling est doublée. La routine scroll3() effectue un scrolling dans le même sens que scroll1(), avec une vitesse double. En diminuant ainsi le nombre d'étapes de scrolling, il est possible d'atteindre une grande vitesse d'exécution.

```
void scroll3()
{
 int i;
 int ligne;

 ligne=0;
 for (i=0; i<200; i++)
   {
     set_ligne(ligne);
     ligne+=2;
   }
}
```

La routine scroll4() effectue un scrolling dans le même sens que scroll2(), mais deux fois plus rapidement.

```
void scroll4()
{
  int i;
  int ligne;

  ligne=199;
  for (i=0; i<200; i++)
    {
      set_ligne(ligne);
      ligne-=2;
    }
}
```

Programme de démonstration

Le programme VSCROLL.C charge deux images *Degas Elite* dans un tampon mémoire, et effectue plusieurs scrollings verticaux à des vitesses différentes. Il fonctionne en haute résolution. Le fichier OSBIND.H est inclu, car il comprend la définition des fonctions Physbase(), Setscreen() et Cconws(). Le fichier MALLOC.H est inclu car il contient la définition de la fonction malloc(). Le fichier MALLOC.H n'existe pas en *Turbo C.* Le fichier STDIO.H s'occupe de la déclaration de malloc().

Ce programme n'utilise que les fonctions du système d'exploitation. C'est donc un programme TOS indépendant du système GEM. Il peut avoir une extension .TTP ou .TOS. Le seul défaut de ce type de programme, c'est la présence permanente d'un curseur clignotant. La fonction eff_curseur() permet d'inhiber l'affichage du curseur. Elle est appelée en début de programme.

```
/*********************************/
/*  SCROLLING VERTICAL           */
/*  EN SOZOBON C                 */
/*********************************/
/*  Programme VSCROLL.C          */
/*********************************/
/*  (C) 1992 Patrick Leclercq    */
/*  (C) 1992 Atari Magazine      */
/*********************************/
#include <STDIO.H>
#include <OSBIND.H>
#include <MALLOC.H>

/***********************************/
/*  VARIABLE GLOBALE CONTENANT     */
/*  L'ADRESSE DE L'ECRAN VIRTUEL */
```

```
/***********************************/
long adr_tampon;

/***********************************/
/*  EFFACEMENT DU CURSEUR TEXTE  */
/***********************************/
void eff_curseur ()
{
  Cconws("\33f");
}

/**********************************/
/*  AFFICHAGE DU CURSEUR TEXTE  */
/**********************************/
void aff_curseur ()
{
  Cconws("\33e");
}

/*****************************/
/*  CHARGEMENT IMAGE DEGAS  */
/*  A UNE ADRESSE MEMOIRE   */
/*****************************/
void load_degas(nom, adr)
char *nom;
long adr;
{
 FILE *fic;
 int  resolution;
 char palette[32];

 fic = fopen(nom, "rb");
 fread (&resolution, 2, 1, fic);
 fread (palette, 32, 1, fic);
 Setpallete (palette);
 fread (adr, 32000, 1, fic);
 fclose (fic);
}

/********************************/
/*  POSITIONNEMENT DE L'ECRAN  */
/*  PHYSIQUE PAR RAPPORT A     */
/*  L'ECRAN VIRTUEL.           */
/********************************/
void set_ligne(l)
int l;
{
  long adr_ecr;

  /* CALCUL ADR. ECRAN PHYSIQUE */
  adr_ecr=adr_tampon+l*80;
  /* CHANGEMENT ADR. ECRAN */
  Setscreen(adr_ecr,adr_ecr,-1);
  /* ATTENTE SYNCHRONISATION */
  Vsync();
}

/*****************************/
/*  SCROLLING VERS LE BAS  */
/*  SCROLL PIXEL PAR PIXEL  */
/*****************************/
void scroll1()
{
 int ligne;

 for (ligne=0; ligne<=400; ligne++)
   set_ligne(ligne);
}

/*****************************/
/*  SCROLLING VERS LE HAUT  */
/*  SCROLL PIXEL PAR PIXEL  */
/*****************************/
void scroll2()
{
 int ligne;

 for (ligne=400; ligne>=0; ligne--)
   set_ligne(ligne);
}

/*******************************/
/*  SCROLLING VERS LE BAS      */
/*  SCROLL TOUS LES 2 LIGNES  */
/*******************************/
void scroll3()
{
  int i;
  int ligne;

  ligne=0;
  for (i=0; i<=200; i++)
    {
      set_ligne(ligne);
      ligne+=2;
    }
}

/*******************************/
/*  SCROLLING VERS LE HAUT     */
/*  SCROLL TOUS LES 2 LIGNES  */
/*******************************/
void scroll4()
{
  int i;
  int ligne;

  ligne=400;
  for (i=0; i<=200; i++)
    {
      set_ligne(ligne);
      ligne-=2;
    }
}

/*******************************/
/*  SCROLLING VERS LE BAS      */
/*  SCROLL TOUS LES 4 LIGNES  */
```

```
/*******************************/
void scroll5()
{
  int i;
  int ligne;

  ligne=0;
  for (i=0; i<=100; i++)
   {
    set_ligne(ligne);
    ligne+=4;
   }
}

/*******************************/
/*  SCROLLING VERS LE HAUT     */
/*  SCROLL TOUS LES 4 LIGNES   */
/*******************************/
void scroll6()
{
  int i;
  int ligne;

  ligne=400;
  for (i=0; i<=100; i++)
   {
    set_ligne(ligne);
    ligne-=4;
   }
}

/*******************************/
/*  SCROLLING VERS LE BAS      */
/*  SCROLL TOUS LES 8 LIGNES   */
/*******************************/
void scroll7()
{
  int i;
  int ligne;

  ligne=0;
  for (i=0; i<=50; i++)
   {
    set_ligne(ligne);
    ligne+=8;
   }
}

/*******************************/
/*  SCROLLING VERS LE HAUT     */
/*  SCROLL TOUS LES 8 LIGNES   */
/*******************************/
void scroll8()
{
  int i;
  int ligne;

  ligne=400;
  for (i=0; i<=50; i++)
   {
    set_ligne(ligne);
    ligne-=8;
   }
}

/*********************************/
/*  DEMONSTRATION DU SCROLLING   */
/*********************************/
void demo_scroll()
{
  scroll1();  getchar();
  scroll2();  getchar();
  scroll3();  getchar();
  scroll4();  getchar();
  scroll5();  getchar();
  scroll6();  getchar();
  scroll7();  getchar();
  scroll8();  getchar();
}

/*************************/
/*  ROUTINE PRINCIPALE   */
/*************************/
void main()
{
  long adr1;
  long adr2;
  long old_ecran;

  /*  STOCKAGE ADR ECRAN DEBUT  */
  old_ecran=Physbase();
  /*  CREATION TAMPON IMAGES  */
  adr_tampon=(long)malloc(64000);
  /*  CALCUL ADRESSES IMAGES  */
  adr1=adr_tampon;
  adr2=adr1+32000;
  /*  CHARGEMENT IMAGES  */
  load_degas("IMAGE1.PI3",adr1);
  load_degas("IMAGE2.PI3",adr2);
  /*  EFFACEMENT CURSEUR TEXTE  */
  eff_curseur();
  /*  EXECUTION SCROLLING  */
  demo_scroll();
  /*  RESTAURATION ECRAN INITIAL  */
  Setscreen(old_ecran,old_ecran,-1);
}
```

Scrolling multirésolution

Pour faire des scrollings dans toutes les résolutions ST, il faut adapter les routines aux différents paramètres de ces résolutions. La mémoire écran basse résolution est organisée en 200 lignes de 160 octets. Chaque ligne contient 320 pixels. La mémoire écran moyenne résolution est organisée en 200 lignes de 160 octets. Chaque ligne contient 640 pixels. La haute résolution est organisée en 400 lignes de 80 octets.

La fonction pos_ecran() positionne l'écran physique sur une ligne quelconque de l'écran virtuel d'adresse adr_tampon. Elle calcule l'a-

dresse de l'écran physique en tenant compte du nombre d'octets par ligne.
Cette information est stockée dans la variable globale nb_octets.

```
void pos_ecran(l)
int l;
{
  long adr_ecr;

  adr_ecr=adr_tampon+l*nb_octets;
  Setscreen(adr_ecr,adr_ecr,-1);
  Vsync();
}
```

La routine init_scroll() utilise la fonction getrez() pour connaître la résolution courante et déterminer le nombre d'octets par ligne. Elle doit être appelée en début de programme.

```
void init_scroll()
{
  int rez;

  rez=getrez();
  if (rez==0) nb_octets=160;
  if (rez==1) nb_octets=160;
  if (rez==2) nb_octets=80;
}
```

Tous les programmes de cet article peuvent être modifiée afin de fonctionner dans toutes les résolutions ST. Attention, il ne faut pas oublier que le nombre de lignes varie selon la résolution (200 lignes en basse et moyenne résolution et 400 lignes en haute résolution).

Scrolling circulaire

Il est possible de réaliser un scrolling circulaire en chargeant deux fois la même image dans l'écran virtuel. Une fois que l'affichage est arrivée en bas de l'écran virtuel, il suffit de le remettre en haut pour que le cycle recommence. Comme c'est un effet cyclique, il faut prévoir un système d'arrêt. La routine exec_scroll() effectue un scrolling du haut vers le bas. Il est possible de stopper le scrolling en pressant sur la barre d'espace. Si la barre d'espace est pressée, exec_scroll() renvoie la valeur 1. Dans le cas contraire, elle renvoie 0.

```
int exec_scroll()
{
  int i;
  int ligne;

  ligne=0;
  for (i=0; i<=100; i++)
   {
     set_ligne(ligne);
     ligne+=4;
     /* TEST TOUCHE <ESPACE> */
     if (inkey()==32) return(1);
   }
  return (0);
}
```

La routine scroll_circulaire() gère le cycle de scrolling. Elle boucle tant que la fonction exec_scroll() renvoie la valeur 0.

```
void scroll_circulaire()
{
  int tst;
  do {
      tst=exec_scroll();
     } while (tst==0);
}
```

Il est possible de simplifier l'écriture de cette fonction en insérant directement l'appel de exec_scroll() dans le test de cyclage. Cela permet d'économiser une variable et un peu de temps d'exécution. C'est une mesure d'optimisation.

```
void scroll_circulaire()
{
  do {} while (exec_scroll()==0);
}
```

La consultation du clavier se fait avec la fonction inkey(). Cette dernière renvoie le code ASCII du caractère frappé au clavier, ou 0 si il n'y a pas de caractères. Elle utilise les fonctions du système d'exploitation du ST pour lire l'état du clavier. Ces fonctions sont définies dans le fichier OSBIND.H. La fonction Cconis() teste si un caractère se trouve dans le tampon clavier. La fonction Crawcin() renvoie le code ASCII du caractère frappé au clavier. Si aucun caractère n'est présent dans le tampon clavier, elle attend. Grâce àCconis(), inkey() n'appelle Crawcin() que si un caractère est effectivement disponible. C'est une routine système que vous pourriez réutiliser dans vos propres programmes.

```
int inkey()
{
  int c;

  /*  INITIALISATION CODE NUL  */
  c=0;
  /*  TEST PRESENCE CARACTERE  */
  if (Cconis()!=0)        /*  LECTURE CARACTERE   */
    c=Crawcin();
  /*  PARAMETRE DE RETOUR  */
  return (c);
}
```

Scrolling commandé par la souris

Il est possible de commander le déplacement du scrolling avec la souris. Il suffit pour cela de lire la position de la souris et d'afficher l'image de l'écran virtuel correspondant à cette position. C'est ce que fait le programme MSCROLL.C. Il utilise les bibliothèques graphiques INITGEM.H et MOUSE.H publiées dans les derniers numéros d'Atari Magazine. Ces fichiers utilisent la bibliothèque graphique *GEMFAST version 1.3*. Vous pouvez la télécharger à partir du serveur Atari sous le nom de FASTG13.TOS.

```
/*********************************/
/*  SCROLLING CONTROLE PAR LA   */
/*  SOURIS - SOZOBON C          */
/*********************************/
/*  Programme MSCROLL.C         */
/*********************************/
/*  (C) 1992 Patrick Leclercq   */
/*  (C) 1992 Atari Magazine     */
/*********************************/
```

```c
#include <STDIO.H>
#include <OSBIND.H>
#include <MALLOC.H>
#include <GEMFAST.H>
#include <INITGEM.H>
#include <MOUSE.H>

/**********************************/
/*  VARIABLE GLOBALE CONTENANT    */
/*  L'ADRESSE DE L'ECRAN VIRTUEL  */
/**********************************/
long adr_tampon;

/****************************/
/*  CHARGEMENT IMAGE DEGAS  */
/****************************/
void load_degas(nom, adr)
char *nom;
long adr;
{
 FILE *fic;
 int resolution;
 char palette[32];

 fic = fopen(nom, "rb");
 fread (&resolution, 2, 1, fic);
 fread (palette, 32, 1, fic);
 Setpallete (palette);
 fread (adr, 32000, 1, fic);
 fclose (fic);
}

/********************************/
/*  POSITIONNEMENT DE L'ECRAN  */
/*  PHYSIQUE PAR RAPPORT A     */
/*  L'ECRAN VIRTUEL.           */
/********************************/
void set_ligne(l)
int l;
{
  long adr_ecr;

  /* CALCUL ADR. ECRAN PHYSIQUE */
  adr_ecr=adr_tampon+l*80;
  /* CHANGEMENT ADR. ECRAN */
  Setscreen(adr_ecr,adr_ecr,-1);
  /* ATTENTE SYNCHRONISATION */
  Vsync();
}

/*********************************/
/*  GESTION DU SCROLLING SOURIS  */
/*********************************/
void scroll_souris()
{
 int ligne;
 int y;
/*  EFFACEMENT SOURIS  */
hidem();
/*  POSITION INITIALE SOURIS  */
ligne=mouseY();
/*  AFFICHAGE IMAGE  */
set_ligne(ligne);
do {
    /* ATTENTE MOUVEMENT SOURIS */
    do {
        /* LECTURE POS SOURIS */
        y=mouseY();
        /* TEST SI FIN SCROLL */
        if (mouseK()!=0) return;
      /* TST SI MODIF POS Y */
      } while(y==ligne);
    ligne=y;
    /* AFFICHAGE IMAGE */
    set_ligne(ligne);
   } while(1);
}

/***********************/
/*  ROUTINE PRINCIPALE  */
/***********************/
void main()
{
 long adr1;
 long adr2;
 long old_ecran;

 /*  INITIALISATION GEM  */
 InitGem();
 /*  STOCKAGE ADR ECRAN DEBUT  */
 old_ecran=Physbase();
 /*  CREATION TAMPON IMAGES  */
 adr_tampon=(long)malloc(64000);
 /*  CALCUL ADRESSES IMAGES  */
 adr1=adr_tampon;
 adr2=adr1+32000;
 /*  CHARGEMENT IMAGES  */
 load_degas("IMAGE1.PI3",adr1);
 load_degas("IMAGE1.PI3",adr2);
 /*  EXECUTION SCROLLING  */
 scroll_souris();
 /*  RESTAURATION ECRAN INITIAL  */
 Setscreen(old_ecran,old_ecran,-1);
 /*  FERMETURE DE GEM  */
 CloseGem();
}
```

Pour finir

Tous les programmes de cet article sont disponibles en téléchargement sur le 3615 ATARI sous le nom de C34.TOS. Ce fichier occupe 7523 octets. Il est auto-décompactable, ce qui évite d'utiliser un utilitaire de décompactage pour récupérer les fichiers.

Patrick Leclercq ■

GFA BASIC
Comment être performant

De nombreux programmeurs ignorent les règles de base de la programmation. Il suffit pourtant de respecter quelques principes simples qui vous éviteront de nombreuses migraines.

Programmation modulaire

Les programmeurs amateurs écrivent souvent des programmes composés de longues séries d'instructions. C'est une grosse erreur. Ces listings sont peu compréhensibles, même par leurs auteurs. Cette méthode les rend très difficiles à débugger et à modifier. Un bon programme doit être écrit sous la forme de petits modules remplissant chacun une tâche bien précise. Cette technique simplifie l'écriture des programmes, diminue le nombre de bugs, et facilite la recherche des bugs présents. De plus, les modules de base peuvent être réutilisés dans d'autres programmes. Selon le langage et la terminologie utilisés, les modules peuvent être appelés procédures, fonctions, routines, sous-routines, etc..

Programmation structurée

La programmation structurée est une technique de programmation qui consiste à écrire des modèles élémentaires qui serviront à écrire des modules plus complexes, et ainsi de suite jusqu'à obtenir le programme final. L'inventeur de la programmation structurée est un universitaire de l'**Ecole polytechnique fédérale**, située en Suisse: Niklaus WIRTH. Les langages du type *Pascal*, *C* ou *GFA Basic* sont des langages structurés, c'est-à-dire qu'ils sont conçus pour faciliter l'écriture de programmes en utilisant les concepts de la programmation structurée.

Création d'un module

L'instruction PROCEDURE permet de créer un module, ou plutôt une procédure pour employer le terme spécifique du *GFA Basic*. Les instructions suivantes sont un exemple de procédure.

```
PROCEDURE boite(px,py,tx,ty,c) ! DEBUT DE LA PROCEDURE
  DEFFILL c                    ! COULEUR RECTANGLE
  px2=px+tx-1                  ! CALCUL POSITION PX2
  py2=py+ty-1                  ! CALCUL POSITION PY2
  PBOX px,py,px2,py2           ! DESSIN RECTANGLE
RETURN                         ! FIN DE LA PROCEDURE
```

L'instruction PBOX permet de dessiner un rectangle de couleur. Le rectangle est défini par ses coordonnées extrêmes (x1,y1,x2,y2). Il serait plus pratique de définir le rectangle par sa position (px,py) et ses dimensions (tx,ty). La procédure boite permet d'afficher un rectangle de couleur c, à partir de sa position et de ses dimensions. Elle s'utilise pratiquement de la même manière qu'une instruction du *GFA Basic*, à ceci près que les paramètres doivent être entre parenthèses.

```
' ****************************************
' *  UTILISATION DE LA PROCEDURE BOITE  *
' ****************************************
' Position X rectangle: 10
' Position Y rectangle: 10
' Largeur rectangle (tx): 65
' Hauteur rectangle (ty): 48
' Couleur rectangle: couleur numéro 2
boite(10,10,65,48,2)
```

Les anciens basics utilisant des numéros de lignes pouvaient appeler des routines grâce à l'instruction GOSUB. Afin de permettre une certaine compatibilité avec la méthode d'utilisation, le *GFA Basic* permet d'appeler une procédure avec l'instruction GOSUB. Cela ne sert pratiquement à rien.

```
GOSUB boite(10,10,65,48,2)
```

L'instruction @ est l'abréviation de l'instruction GOSUB. Elle peut être utilisée pour appeler n'importe quelle procédure.
C'est la syntaxe qu'utilise le plus souvent l'auteur de cet article, de manière à se servir de la même syntaxe d'appel pour les procédures et les fonctions.

```
@boite(10,10,65,48,2)
```

Création d'une fonction

Les fonctions sont une variante des procédures. Elles exécutent une tâche quelconque, puis renvoient une valeur au programme. Cette valeur peut être le résultat d'un calcul, ou une indication sur la bonne exécution de la fonction.
La fonction prix_ttc est un exemple de fonction. Elle calcule un prix TTC à partir d'un prix HT. L'instruction RETURN renvoie la valeur de retour de la fonction. L'instruction ENDFUNC signale la fin de la fonction.

```
FUNCTION prix_ttc(ht)  ! DEBUT DE LA FONCTION
  ttc=ht*1.186         ! CALCUL DU PRIX TTC
  RETURN ttc           ! RENVOI DE LA VALEUR FONCTION
ENDFUNC                ! FIN DE LA FONCTION
```

Les instructions suivantes vous montrent comment utiliser une fonction. Il est indispensable de mettre le symbole @ devant le nom de la fonction, sinon le *Gfa Basic* va comprendre que vous voulez lire le contenu du tableau prix_ttc(). Et comme ce tableau n'existe pas, il y aura erreur d'exécution.

```
prix=@prix_ttc(120)       ! CALCUL PRIX TTC
PRINT "Prix TTC: ";prix   ! AFFICHAGE PRIX TTC
```

Création d'une procédure à partir d'autres procédures

Il est possible d'écrire une procédure à partir d'autres procédures, ou de fonctions. La procédure draw_cadres dessine plusieurs cadres sur l'écran, en utilisant la procédure boite définie plus haut.

```
PROCEDURE draw_cadres
  @boite(10,10,32,32,4)
  @boite(60,60,12,12,4)
RETURN
```

Cette possibilité de créer des procédures à partir d'autres procédures est indispensable pour utiliser les principes de la programmation structurée.

Les variables locales

Chaque procédure doit exécuter une tâche bien précise. Pour cela, elle effectue un traitement interne. Ce traitement ne doit pas affecter les autres procédures. Le fonctionnement interne d'une procédure doit être indépendant des autres procédures. Pour ce faire, il faut utiliser des variables qui n'appartiennent qu'à la procédure: les variables locales. L'instruction LOCAL permet de définir des variables locales qui n'existent qu'à l'intérieur d'une procédure ou d'une fonction. Le programme suivant est un exemple de procédures utilisant des variables locales.

```
PROCEDURE proc1
  LOCAL i,r
  '
  r=0
  FOR i=1 TO 10
    INC r
    @proc2(r)
  NEXT i
RETURN

PROCEDURE proc2(v)
  LOCAL i
  '
  i=2*3
  PRINT i
RETURN
```

Quelque soient les valeurs écrites dans la variable i de la procédure proc1, elles n'influent pas sur la valeur de la variable i de la procédure proc2 Les variables locales permettent de créer de véritables «boîtes noires» dont le fonctionnement interne des unes n'influe pas sur les autres. Si vous n'utilisez pas les variables locales, il faut tenir des listes de variables utilisées (i0, i1, i2, i3, etc.). C'est un travail énorme qui comporte de nombreux risques d'erreurs.

Les variables locales peuvent être de n'importe quel type (réels, entiers court, entiers long, chaînes de caractères, etc.) En revanche, il est impossible de créer des tableaux locaux. C'est dommage, car ce serait pratique dans certains cas. Le *Pascal* et le *Langage C* permettent de créer des tableaux locaux.

L'utilisation des variables locales facilite aussi la réutilisation des procédures dans d'autres programmes. Sans variables locales, l'intégration d'une routine standard dans un programme est un véritable cauchemar, car il faut vérifier si toutes les variables de la procédure ne risquent pas de perturber les variables du programme. Avec les variables locales, il suffit d'insérer le code source de la fonction dans le programme. C'est un grand progrès par rapport aux anciens basics.

Ecriture d'une procédure

Avant tout, il faut définir ce que doit faire la routine, et comment elle doit le faire. Il faut penser à la manière dont elle communique avec l'extérieur (paramètres d'entrée et de sortie), ainsi que la gestion des erreurs. La routine doit être relativement courte, et n'exécuter qu'une seule tâche. Une bonne technique est d'écrire sur papier, tout ce que doit faire la procédure.

L'idéal serait que la totalité du code source d'un module soit visible sur l'écran. En pratique, c'est impossible à réaliser sans faire des découpages artificiels dans la structure du programme. Essayez d'écrire des routines ne dépassant pas la taille de deux ou trois pages-écrans.

Structures de contrôle

Les structures de contrôle sont des instructions qui permettent de contrôler l'exécution du programme en fonction des besoins du traitement. Les structures les plus pratiques sont la boucle FOR-NEXT et la boucle DO-LOOP. La boucle FOR-NEXT permet de répéter une séquence d'instructions pendant un nombre déterminé de fois.

```
FOR i=1 TO 20
  PRINT i
NEXT i
```

La boucle DO-LOOP permet d'exécuter une séquence d'instructions jusqu'à ce qu'une condition de sortie soit réalisée. Le test de sortie est réalisé par l'instruction EXIT IF.

```
DO
  ...
  EXIT IF sortie=1
LOOP
```

C'est une structure de contrôle très pratique, puisque le test de sortie peut être situé à n'importe quel endroit de la boucle. La procédure

attclavier est un exemple de procédure utilisant une boucle DO-LOOP. Elle bloque l'exécution du programme tant que l'utilisateur ne presse pas sur une touche du clavier.

```
PROCEDURE attclavier
  DO
    EXIT IF INKEY$<>""
  LOOP
RETURN
```

La procédure saisie_code est une routine de saisie d'un code secret. Chaque fois que l'utilisateur frappe sur une touche, la routine affiche le symbole ".". La fin du code est signalée par la pression sur la touche <RETURN>. La chaîne de caractères contenant le code secret est mémorisée dans la variable globale m$.

Cette routine utilise deux boucles DO-LOOP imbriquées l'une dans l'autre. La boucle principale contrôle le processus de saisie du code. La boucle interne bloque l'exécution du programme tant que l'utilisateur ne presse pas sur une touche. Si vous voulez utiliser cette routine dans un programme, il faudra ajouter la possibilité d'effacer les caractères du code. Utilisez pour cela la touche <Backspace> afin que l'utilisateur ne soit pas désorienté par une touche d'effacement inhabituelle.

```
PROCEDURE saisie_code
 LOCAL c$
 '
 m$=""                      ! INITIALISATION CODE SECRET
 DO                         ! DEBUT BOUCLE DE SAISIE CODE
   DO                       ! DEBUT BOUCLE ATTENTE CAR.
     c$=INKEY$              ! LECTURE CLAVIER
     EXIT IF c$<>""         ! TEST SI PRESENCE CAR.
   LOOP                     ! FIN BOUCLE ATTENTE CAR.
   EXIT IF c$=CHR$(13)      ! SORTIE SI FRAPPE SUR <RETURN>
   m$=m$+c$                 ! MEMORISATION DU CARACTERE
   PRINT ".";               ! AFFICHAGE SYMBOLE "."
 LOOP                       ! FIN BOUCLE DE SAISIE
RETURN
```

Il existe d'autres structures de contrôle (REPEAT-UNTIL, WHILE-WEND, DO-WHILE, DO-UNTIL, etc.). Elles correspondent à des cas particuliers pouvant être reproduits avec la boucle DO-LOOP.

Système de choix multiple

L'instruction SELECT permet de gérer facilement les tests multiples sans utiliser une longue série d'instructions IF.

```
IF n=1
  @exec_proc1
ENDIF
IF n=2
  @exec_proc2
ENDIF
IF (n=3) AND (n=4)
  @exec_proc3
ENDIF
```

La routine écrite avec la structure SELECT-CASE est plus lisible, et plus facilement modifiable.

```
SELECT n
CASE 1
  @exec_proc1
CASE 2
  @exec_proc2
CASE 3,4
  @exec_proc3
ENDSELECT
```

Cet exemple ne montre pas toutes les possibilités de SELECT. Pour toutes les connaître, référez-vous au manuel du *GFA Basic*.

Non à l'instruction GOTO

L'instruction GOTO est une horreur qui nous provient de la préhistoire de l'informatique. Elle permet de créer des programmes extraordinairement illisibles et quasi-impossibles à débugger. Elle était indispensable avec les anciens basics utilisant des numéros de lignes et ne disposant pas de structures de contrôle du type DO-LOOP. Elle sert surtout à effectuer des branchements entre blocs d'instructions, opération qui se fait nettement plus efficacement en utilisant les structures de contrôle et les procédures. Des spécialistes de l'informatique ont prouvé de manière théorique qu'il est possible d'écrire n'importe quel programme sans utiliser l'instruction GOTO. Aucun des nombreux programmes publiés dans **Atari Magazine** par l'auteur de ces lignes n'utilisent GOTO. Si vous voulez écrire des programmes performants, oubliez jusqu'à l'existence même de GOTO.

Utilisation de noms significatifs

Il faut utiliser autant que possible des noms significatifs pour les procédures et les variables. Si vous avez une variable contenant le nombre d'images stocké en mémoire, il est plus parlant de l'appeler nb_images que n ou n1. De la même manière, le nom des routines doit être le plus précis possible. En procédant de la sorte, les listings sources deviennent beaucoup plus lisible. Cette méthode s'appelle l'auto-documentation. Les avantages de cette technique compensent largement la perte de temps causée par la saisie d'identificateurs de grandes dimensions.

```
PROCEDURE main
  @initialisation
  @lecture_fichier("INFOS.DTA")
  @dessin_ecran
  @exec_jeu
  @desinitialisation
RETURN
```

Utilisation des commentaires

Il est très important de commenter les programmes. Tous devraient commencer par un en-tête énonçant le nom du programme, ce qu'il fait, le nom du programmeur et la version du langage utilisé (*GFA Basic 2.0*, *3.0*, *3.3*, *3.5* ou *3.5E*).

```
' ******************************
' *  DESCRIPTION DU PROGRAMME   *
' ******************************
' *  (C) 1992 Patrick Leclercq  *
' *  (C) 1992 Atari Magazine    *
' ******************************
' *  Langage GFA Basic 3.5E     *
' ******************************
```

La mise au point d'un programme nécessite de nombreuses versions. Afin de les différencier, il est possible de les numéroter et de noter ce numéro en début de programme. Cela permet d'identifier rapidement un fichier. Les versions principales des programmes doivent être référencés par des numéros entiers (version 1, 2, etc.) Les améliorations mineures des logiciels doivent être référencées par un numéro secondaire (version 1.1, 1.2, voire même 1.121).

```
' ****************
' *  VERSION 1.2  *
' ****************
```

Si possible, toutes les routines doivent avoir un en-tête décrivant la fonction de la routine. Si la syntaxe d'utilisation de la routine est complexe, il est préférable de rappeler les détails de cette syntaxe.

```
' ***************************************
' *  CHARGEMENT D'UNE IMAGE DEGAS ELITE  *
' *  A UNE ADRESSE QUELCONQUE.           *
' ***************************************
' *  nom$: Nom de l'image à charger      *
' *  adr%: Adresse mémoire               *
' ***************************************
PROCEDURE charge_degas(nom$,adr%)
..
RETURN
```

Si la routine intègre un système de contrôle d'erreurs, il faut noter le fonctionnement de ce système en commentaires. Prenons par exemple une routine cherche_chaine qui cherche l'emplacement d'une chaîne de caractères stockée dans un tableau. Elle renvoie l'indice du tableau où se trouve la chaîne. En cas de recherche négative, elle renvoie la valeur -1. Ce comportement est suffisamment complexe pour être noté dans le corps du programme.

```
' ******************************************
' *  RECHERCHE DE LA POSITION D'UNE CHAINE  *
' *  DE CARACTERES DANS LE TABLEAU nom$().  *
' ******************************************
' *  Cette fonction renvoie la position de  *
' *  la chaîne c$ dans le tableau nom$().   *
' *  En cas de recherche négative, elle     *
' *  renvoie la valeur -1.                  *
' ******************************************
FUNCTION cherche_chaine(c$)
  LOCAL i          ! INDICE DE BOUCLE
  LOCAL rep        ! NUMERO DE CHAINE
  '
  rep=-1           ! INIT. REPONSE NEGATIVE
  FOR i=1 TO 100   ! BOUCLE DE TEST
    IF nom$(i)=c$  ! COMPARAISON DE CHAINES
      rep=i        ! MEMORISATION POSITION CHAINE
    ENDIF
  NEXT i           ! FIN BOUCLE DE TEST
  RETURN rep       ! RENVOIT PARAMETRE DE LA FONCTION
ENDFUNC
```

Les commentaires de cette routine sont très complets, trop complets pour un programme réel. Il est impossible de commenter systématiquement toutes les instructions sans perdre beaucoup de temps Vous devez trouver un style de commentaires qui fournit des indications précieuses sur le fonctionnement du programme, tout en évitant de donner des informations qui peuvent se déduire facilement de l'examen du listing. La plupart des programmes d'**Atari Magazine** sont sur-commentés pour des raisons pédagogiques.

De nombreux programmeurs limitent les commentaires au minimum, car ils ont appris à travailler sur des machines dotées d'une faible mémoire centrale, et d'une mémoire de masse peu performante. Ces limitations ne sont plus valables avec une machine comme le ST qui possède une mémoire centrale de grande dimension et un lecteur de disquette.

Communication entre routines

Les routines peuvent communiquer entre elles par des messages. Le type de messages le plus simple est une valeur numérique. Ces commandes peuvent être des informations, ou des codes d'erreurs. Prenons par exemple la procédure exec_programme qui exécute les commandes de l'utilisateur. Elle utilise la fonction att_ordre qui renvoie le numéro de la commande demandée par l'utilisateur. La procédure exec_ordre s'occupe d'appeler les routines qui correspondent aux demandes utilisateurs. Le numéro de message -1 correspondant à la commande "Quitter programme".

```
PROCEDURE exec_programme
  LOCAL com
  '
  DO
    com=@att_ordre    ! ATTENTE COMMANDE UTILISATEUR
    EXIT IF com=-1    ! TEST SI SORTIE
    @exec_ordre(com)  ! EXECUTION COMMANDE
  LOOP
RETURN
```

Gestion des erreurs

Etant donné que les programmes ne fonctionnent pas dans un monde parfait, mais dans le monde réel, il peut se produire des erreurs d'exécution. Ce peut être une tentative de lire le contenu d'un fichier inexistant, l'impression sur une imprimante absente, l'écriture sur une disquette pleine, etc. Les modules doivent être capables de détecter de telles erreurs et d'informer le programme qu'une tâche n'a pu être exécutée pour telle ou telle raison.

La fonction exec_sauvegarde(fic$) sauve un ensemble de données dans le fichier de nom fic$. Si l'opération se passe correcte-

ment, elle renvoie la valeur 1. Dans le cas contraire, elle renvoie la valeur 0. La procédure sauvegarde_donnes contrôle la bonne exécution de la sauvegarde et affiche une boîte d'alerte en cas d'erreur. L'utilisateur peut alors changer de disquette pour recommencer l'opération de sauvegarde, ou stopper l'opération. Remarque: le programme peut quitter la boucle DO-LOOP à deux endroits différents.

```
FUNCTION exec_sauvegarde(fic$)
..
ENDFUNC

PROCEDURE sauvegarde_donnees(fic$)
  LOCAL err$,op$
  LOCAL test,b
  '
  DO
    test=@exec_sauvegarde(fic$)  ! EXECUTION SAUVEGARDE
    EXIT IF test=1               ! FIN SI EXECUTION OK
    err$="Erreur de sauvegarde"  ! MESSAGE ALERTE
    op$="Recommencer|Abandonner" ! MESSAGE ALERTE
    ALERT 0,err$,1,op$,b         ! AFF. BOITE D'ALERTE
    EXIT IF b=2                  ! TEST SI REFAIRE
                                 ! SAUVEGARDE
  LOOP
RETURN
```

La nature des erreurs doit être affichée avec le maximum d'informations. Dans le cas de la sauvegarde de données sur disque, les erreurs peuvent être dues à l'absence de disquette dans le lecteur, ou la présence d'une disquette pleine. Pour tenir compte de ceci, il faut modifier la routine exec_sauvegarde pour qu'elle retourne la valeur 1 en cas de sauvegarde réussie, la valeur 2 en cas de disquette pleine, et la valeur 3 si aucune disquette n'est présente dans le lecteur. La nouvelle version de sauvegarde_donnees tient compte de ces valeurs pour afficher des messages d'erreurs spécifiques.

```
FUNCTION exec_sauvegarde(fic$)
..
ENDFUNC

PROCEDURE sauvegarde_donnees(fic$)
  LOCAL err$,op$
  LOCAL test,b
  '
  DO
    test=@exec_sauvegarde(fic$)  ! EXECUTION SAUVEGARDE
    EXIT IF test=1               ! FIN SI EXECUTION OK
    IF test=2                    ! TEST ERREUR 2
      err$="Disquette pleine"    ! MESSAGE ERREUR 2
    ENDIF
    IF test=3                    ! TEST ERREUR 3
      err$="Pas de disquette"    ! MESSAGE ERREUR 3
    ENDIF
    op$="Recommencer|Abandonner" ! MESSAGE ALERTE
    ALERT 0,err$,1,op$,b         ! AFF. BOITE D'ALERTE
    EXIT IF b=2                  ! TEST SI REFAIRE
                                 ! SAUVEGARDE
  LOOP
RETURN
```

Mise au point des procédures

Une fois l'écriture d'une procédure achevée, il faut la tester. Les tests se font en vérifiant le comportement de la procédure dans tous les cas possibles. Les erreurs peuvent être de deux types: les erreurs de logique et les bugs. Les erreurs de logique proviennent d'une mauvaise conception de la procédure. Nous avons traité des bugs dans un article récent.
Le test d'une procédure écrite à partir d'autres procédures peut poser des problèmes.
Pour les résoudre, il suffit de remplacer ces procédures par des procédures vides ou minimales.
Par exemple, pour tester une routine de traitement d'erreurs, il suffit de remplacer la routine de traitement par une routine renvoyant systématiquement un code d'erreurs.

```
' ********************************
' *  ROUTINE DE TEST RENVOYANT  *
' *  SYSTEMATIQUEMENT UN CODE   *
' *  D'ERREUR.                  *
' ********************************
FUNCTION exec_fonction
  RETURN 2
ENDFUNC

PROCEDURE gestion_fonction
  LOCAL test
  '
  DO
    test=@exec_fonction ! APPEL ROUTINE DE TRAITEMENT
    EXIT IF test=1
    ....                ! GESTION ERREUR
  LOOP
RETURN
```

Optimisation

Optimiser un programme, c'est le rendre plus performant. Cela peut être fait en le rendant plus rapide, ou plus économique en mémoire. C'est une erreur que de vouloir optimiser un programme à tout prix. Il ne faut optimiser que les parties critiques du programme, surtout les routines graphiques et les routines de calculs.
L'optimisation peut se faire en trouvant un nouvel algorithme, ou en «triturant» le listing dans tous les sens de manière à gagner quelques milli-secondes ici et là.
Pour optimiser une routine, il faut limiter le nombre de variables intermédiaires, utiliser de préférence des variables entières au lieu de variables réelles, employer les structures de contrôle les plus rapides, etc.

```
' **********************
' *  ROUTINE INITIALE  *
' **********************
FOR i=1 TO 100
```

```
  @proc(i)
NEXT i

' ***********************
' *  ROUTINE OPTIMISEE  *
' ***********************
FOR i%=1 TO 100
  @proc(i%)
NEXT i%
```

Généralement, les listings des routines optimisées ne sont pas très lisibles. Il faut donc écrire de nombreux commentaires pour comprendre ce qui se passe dans ces routines. Il est très frustrant de réécrire une routine dont on n'arrive plus à comprendre le fonctionnement de la superbe «bidouille» qui fait gagner tant de temps. Cela arrive fréquemment lorsque les programmes doivent être modifiés quelques mois après qu'ils aient été écrits.

L'instruction TIMER donne le temps écoulé en milli-secondes depuis l'allumage du ST. Vous pouvez l'utiliser pour mesurer la vitesse d'exécution d'une routine. Cela permet de faire des comparaisons entre divers routines. Pour ce faire, il faut mémoriser l'état de la variable TIMER avant et après l'exécution de la routine.

```
t1%=TIMER        ! MESURE TEMPS INITIAL
@exec_routine    ! EXECUTION ROUTINE
t2%=TIMER        ! MESURE TEMPS TERMINAL
t3%=t2%-t1%      ! CALCUL TEMPS EXECUTION
PRINT t3%        ! AFFICHAGE TEMPS EXECUTION
```

Si le temps d'exécution des routines est trop rapide pour que la mesure puisse être valable, il faut exécuter plusieurs fois la routine et mesurer le temps globale.

```
t1%=TIMER          ! MESURE TEMPS INITIAL
FOR i%=1 TO 1000   ! BOUCLE DE MESURE
  @exec_routine    ! EXECUTION ROUTINE
NEXT i%            ! FIN BOUCLE DE MESURE
t2%=TIMER          ! MESURE TEMPS TERMINAL
t3%=t2%-t1%        ! CALCUL TEMPS EXECUTION
PRINT t3%          ! AFFICHAGE TEMPS EXECUTION
```

Le traitement de la boucle prend un peu de temps. Dans certains cas, cela peut interférer sur la mesure finale. Pour remédier à ça, il faut mesurer le temps d'exécution de la boucle à vide, et retirer ce temps de la mesure finale.

```
' **********************************
' *  MESURE DU TEMPS D'EXECUTION   *
' *  DE LA BOUCLE A VIDE           *
' **********************************
t1%=TIMER          ! MESURE TEMPS INITIAL
FOR i%=1 TO 1000   ! DEBUT BOUCLE VIDE
NEXT i%            ! FIN BOUCLE VIDE
t2%=TIMER          ! MESURE TEMPS TERMINAL
t0%=t2%-t1%        ! CALCUL TEMPS BOUCLE VIDE
' **********************************
' *  MESURE DU TEMPS D'EXECUTION   *
' *  DE LA ROUTINE                 *
' **********************************
t1%=TIMER          ! MESURE TEMPS INITIAL
FOR i%=1 TO 1000   ! DEBUT BOUCLE MESURE
  @exec_routine    ! EXECUTION ROUTINE
NEXT i%            ! FIN BOUCLE MESURE
t2%=TIMER          ! MESURE TEMPS TERMINAL
t3%=t2%-t1%        ! CALCUL TEMPS EXECUTION ROUTINE
t3%=t3-t0%         ! RETIRER TEMPS BOUCLE A VIDE
PRINT t3%          ! AFFICHER TEMPS EXECUTION
```

Conception de l'interface utilisateur

L'interface utilisateur d'un programme doit être particulièrement soignée, puisque c'est uniquement par son intermédiaire que l'utilisateur se sert du programme. Une bonne interface doit être simple et intuitive, c'est-à-dire que l'utilisateur doit pouvoir se servir du programme sans consulter la documentation pour connaître les commandes élémentaires. L'utilisation de la souris, des menus déroulants et des écrans graphiques permet d'écrire des programmes ergonomiques, mais ne justifie pas sa facilité d'utilisation.

Pour concevoir une interface, il faut se mettre dans la tête de l'utilisateur et penser comme lui. Chaque écran doit être testé de nombreuses fois. Il ne faut pas hésiter à réaliser et à tester plusieurs versions de la même interface. C'est certainement la chose la plus difficile et la plus longue à faire dans un programme. Beaucoup de programmeurs consacrent tout leur temps à la mise au point de routines ultra optimisés et ultra performantes sans trop s'occuper de l'interface utilisateur. C'est pourquoi il existe tant de programmes puissants, mais difficile d'emploi.

Les tâches que doit remplir une interface sont l'attente des ordres de l'utilisateur, la saisie de données et la fourniture d'informations. De plus, l'interface doit informer l'utilisateur des éventuelles erreurs d'exécution.

Pour finir

Mettez ces conseils en pratique pour écrire de beaux programmes. Envoyez-les nous, nous mettrons les meilleurs en téléchargement. Etant donné que les routines de cet article ne sont pas des programmes fonctionnels, mais de petits exemples isolés, nous ne les avons pas mis en téléchargement.

Patrick Leclercq

PREMIERS PAS EN OMIKRON
Les masques

Après les transferts de blocs au moyen de l'instruction BITBLT, cherchons à expliquer ce qui occasionne des nuits blanches à de nombreux programmeurs de logiciels de DAO, à savoir les masques.

Nous avez pu constater dans notre dernier article que nous n'avons guère insisté sur les modes logiques de l'instruction BITBLT. Cette démarche était volontaire. La logique booléenne est très rébarbative sans exemples concrets.

Qu'est-ce qu'une opération booléenne?

Une opération booléenne (du mathématicien anglais George Boole) est une opération logique s'appuyant sur le système binaire cher à nos ordinateurs. La représentation mathématique de ce système se fait en base deux, c'est-à-dire à l'aide des chiffres 0 et 1. Il existe quatre opérations logiques fondamentales: AND (ET), OR (OU inclusif), XOR (OU exclusif) et NOT (Négation). Il est très facile d'assimiler ces opérations logiques à des idées concrètes: 1 = vrai, allumé, ami, etc., et 0 = faux, éteint, ennemi, etc. Pour chaque opération logique, on crée une table de vérité.

Précisons que le OR est dit inclusif car il inclut la condition 1 OU 1 = 1. A l'inverse, le XOR est dit exclusif car il exclut cette condition («c'est l'un ou l'autre, pas les deux»). De plus, vous trouverez dans le manuel de références du *Basic Omikron* d'autres opérations logiques qui sont des combinaisons des fonctions de base. Par exemple, un NAND va être un NOT combiné avec un AND...

Et à l'écran, qu'est-ce que cela donne?

L'instruction BITBLT utilise le code logique pour effectuer l'opération correspondante entre tous les points du bloc et les points de la zone de l'écran sur laquelle il se pose. Ainsi, avec le code 1 (AND), seuls les points qui seront allumés dans le bloc et sur l'écran le seront au résultat final. Avec le code 7 (OR), il suffit qu'un des deux points soit allumé pour que le point résultant de l'opération logique le soit à son tour. Il s'en suit un effet de transparence...

Qu'est-ce qu'un masque?

Si vous avez saisi et utilisé le programme de démonstration de notre précédent article, vous avez pu vous rendre compte qu'il n'était pas possible d'obtenir un véritable bloc comme ceux que l'on peut utiliser dans des logiciels de DAO. La raison en est simple. On découpe toujours une surface rectangulaire autour du motif qui nous intéresse véritablement. Il en résulte un affichage altéré par ce débordement... Il y a bien la solution d'utiliser le mode 7 (OR) correspondant à un effet de transparence, mais le bloc entier sera transparent, et ce n'est pas ce que l'on cherche. En fait, on cherche à ce que la forme contenue dans notre bloc soit opaque, et que sa périphérie soit transparente. Comment est-ce possible?

La solution réside dans la constitution d'un masque, c'est-à-dire la création d'un «bloc de fond» sur lequel va se poser notre bloc découpé. En d'autres termes, il faut calculer une «ombre chinoise» du motif découpé que l'on placera avec un mode logique bien déterminé et sur lequel on juxtaposera notre bloc d'origine.

La construction d'un masque

La construction d'un masque est le résultat d'une démarche logique assez complexe, plutôt que de l'énoncer, nous allons l'aborder sous forme d'un exemple pratique, avec une illustration graphique à l'appui. A partir de la découpe de notre bloc d'origine, nous allons chercher à obtenir un deuxième bloc (le masque) qui, une fois posé sur l'écran, formera une «ombre chinoise» de la couleur du fond avec la forme contenue dans le bloc (un cercle dans notre exemple). Tout ce qui est extérieur au cercle devra être transparent.

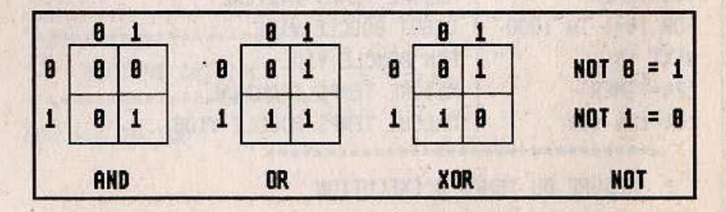
Etape n°1: On découpe un bloc rectangulaire autour de notre cercle. Le bloc ne se limite pas qu'au cercle, mais à l'ensemble de la découpe rectangulaire. C'est ici que réside la difficulté.

Etape n°2: On dessine une boîte (BOX) ayant les coordonnées du bloc découpé afin de «matérialiser» sa limite. Nous verrons dans le programme de démonstration qu'il est préférable de prendre un ou deux pixels de marge.

Etape n°3: On remplit (FILL) la boîte ainsi délimitée. Le remplissage «contourne» notre cercle et se limite aux coordonnées de la boîte.

Etape n°4: On effectue un NOT avec le bloc d'origine, tous les pixels sont donc inversés. Puis, on effectue un AND entre le bloc ainsi obtenu et le bloc obtenu à l'étape n°3. Ainsi, seuls les points à la périphérie du cercle sont allumés dans les deux blocs. Il en résulte une ombre chinoise inversée du cercle.

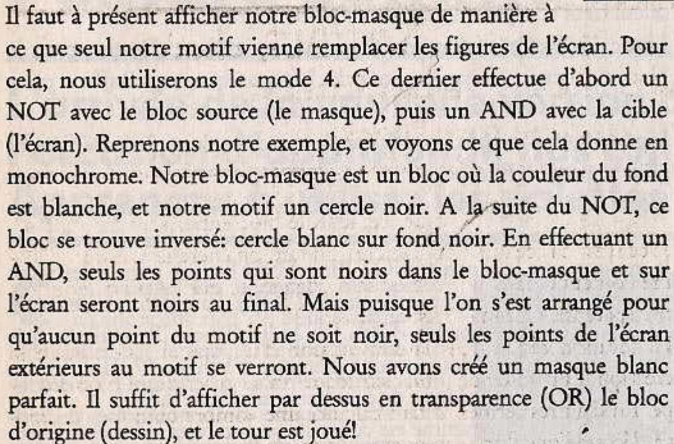
Etape n°5: On inverse de nouveau le bloc obtenu pour obtenir le masque définitif.



A l'issue de cette manipulation, notre masque est donc un bloc de la même taille que notre bloc d'origine, dont le contenu est notre motif en ombre chinoise.


Comment éviter ceci... ...tout en n'utilisant pas la transparence... ... et obtenir ceci !

L'affichage du masque et du bloc



Il faut à présent afficher notre bloc-masque de manière à ce que seul notre motif vienne remplacer les figures de l'écran. Pour cela, nous utiliserons le mode 4. Ce dernier effectue d'abord un NOT avec le bloc source (le masque), puis un AND avec la cible (l'écran). Reprenons notre exemple, et voyons ce que cela donne en monochrome. Notre bloc-masque est un bloc où la couleur du fond est blanche, et notre motif un cercle noir. A la suite du NOT, ce bloc se trouve inversé: cercle blanc sur fond noir. En effectuant un AND, seuls les points qui sont noirs dans le bloc-masque et sur l'écran seront noirs au final. Mais puisque l'on s'est arrangé pour qu'aucun point du motif ne soit noir, seuls les points de l'écran extérieurs au motif se verront. Nous avons créé un masque blanc parfait. Il suffit d'afficher par dessus en transparence (OR) le bloc d'origine (dessin), et le tour est joué!

Une autre astuce consiste également à créer un «masque évidé» en décalant de un ou deux pixels dans les huit directions le masque une fois calculé. Nous aurons donc un masque «débordant» sous le dessin, et ceci pour permettre de voir le tracé du dessin lorsqu'on le déplace sur une même couleur que sa couleur de fond.

Le listing

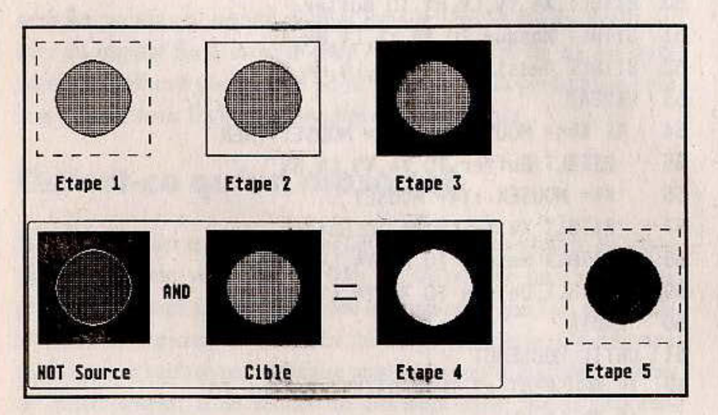
Voici un listing de démonstration qui reprend en partie le travail qui a déjà été accompli le mois dernier. Le détail des commentaires se portera uniquement sur le calcul des masques (voir Atari Magazine n°32). Cependant, une «coquille» d'impression peut toujours se glisser. Tous les listings sont testés sérieusement avant d'être publiés.

C'est pourquoi, nous ne saurions trop vous conseiller de télécharger les listings sur le 3615 ATARI, à partir du moment où vous estimez que sa longueur risque d'entraîner des erreurs de saisie. Pour notre part, nous allons éclaircir les listings, c'est à dire mettre moins d'instructions par ligne, de manière à en faciliter la lecture et la saisie.

```
'
CLEAR 100000: XBIOS (Rez%,4)
IF Rez%=0 THEN Lc%=320:Hc%=200:Np%=4
IF Rez%=1 THEN Lc%=640:Hc%=200:Np%=2
IF Rez%=2 THEN Lc%=640:Hc%=400:Np%=1
CLIP 0,0,Lc%,Hc%: MODE =1
CLS : PRINT CHR$(27);"f";
LINE STYLE =1: LINE WIDTH = 1
'
Ecran= MEMORY(32256)+255 AND $FFF00
Phys= LPEEK($44E)
'
FILL COLOR =2: FILL STYLE =1,1
PBOX 20,20,Lc%\5,Hc%\5
FILL COLOR =3: FILL STYLE =2,4
PBOX Lc%\2,20,Lc%\4,Hc%\4
FILL COLOR =1: FILL STYLE =2,20
PCIRCLE Lc%\2,Hc%\2+20,Hc%\5
FILL STYLE =1,1: MOUSEON
'
REPEAT
 REPEAT
  Xo%= MOUSEX :Yo%= MOUSEY
 UNTIL MOUSEBUT =1
 MODE =3: LINE STYLE =5: MOUSEOFF
 WHILE MOUSEBUT <>0
  X%= MOUSEX :Y%= MOUSEY
  IF X%<> MOUSEX OR Y%<> MOUSEY THEN
   L%= ABS(Xo%-X%):H%= ABS(Yo%-Y%)
   BOX Xo%,Yo%,L%,H%: BOX Xo%,Yo%,L%,H%
  ELSE
   X%= MOUSEX :Y%= MOUSEY
   L%= ABS(Xo%-X%):H%= ABS(Yo%-Y%)
   BOX Xo%,Yo%,L%,H%
   WHILE X%=MOUSEX AND Y%=MOUSEY AND
   MOUSEBUT<>0
   WEND
   BOX Xo%,Yo%,L%,H%
  ENDIF
 WEND
 LINE STYLE =1: MODE =1
 Taille=6+(L%+15) SHR 4*H%*2*Np%
 Dessin= MEMORY(Taille):Masque= MEMORY(Taille)
 Buffer= MEMORY(Taille)
 BITBLT Xo%,Yo%,L%,H% TO Dessin
 '
 MOUSEON
 FORM_ALERT (1,"[0][Masque :]
 [Plein|Détouré]",C%)
 MOUSEOFF
 IF C%=1 THEN Full ELSE Around
 '
 BITBLT X%,Y%,L%,H% TO Buffer
 BITBLT Masque TO X%,Y%,L%,H%,4
 BITBLT Dessin TO X%,Y%,L%,H%,6
 REPEAT
  IF X%<> MOUSEX OR Y%<> MOUSEY THEN
   BITBLT Buffer TO X%,Y%,L%,H%
   X%= MOUSEX :Y%= MOUSEY
   BITBLT X%,Y%,L%,H% TO Buffer
   BITBLT Masque TO X%,Y%,L%,H%,4
   BITBLT Dessin TO X%,Y%,L%,H%,7
  ENDIF
 UNTIL MOUSEBUT
 IF MOUSEBUT =2 THEN BITBLT Buffer TO
 X%,Y%,L%,H%
```

```
63  MOUSEON : WHILE MOUSEBUT : WEND
64  FORM_ALERT (1,"[0][Et puis...]
    [Encore|Fin]",X%)
65  FRE (Buffer): FRE (Dessin): FRE (Masque)
66 UNTIL X%=2
67 IF COMPILER THEN SYSTEM ELSE EDIT
68 '
69 ' Calcul d'un masque plein.
70 '
71 DEF PROC Full
72  IPL 7
73  XBIOS (,5,L Ecran,L Phys,-1): CLS
74  BITBLT Dessin TO Xo%,Yo%,L%,H%
75  BOX Xo%-2,Yo%-2,L%+4,H%+4
76  FILL COLOR =1: FILL Xo%-1,Yo%-1,-1
77  BITBLT Dessin TO Xo%,Yo%,L%,H%,4
78  BITBLT Dessin TO Xo%,Yo%,L%,H%,10
79  BITBLT Xo%,Yo%,L%,H% TO Masque
80  XBIOS (,5,L Phys,L Phys,-1)
81  IPL 3
82 RETURN
83 '
84 ' Calcul d'un masque détouré.
85 '
86 DEF PROC Around
87  Full: IPL 7
88  XBIOS (,5,L Ecran,L Phys,-1)
89  BITBLT Masque TO Xo%-1,Yo%,L%,H%,7
90  BITBLT Masque TO Xo%-1,Yo%-1,L%,H%,7
91  BITBLT Masque TO Xo%,Yo%-1,L%,H%,7
92  BITBLT Masque TO Xo%+1,Yo%-1,L%,H%,7
93  BITBLT Masque TO Xo%+1,Yo%,L%,H%,7
94  BITBLT Masque TO Xo%+1,Yo%+1,L%,H%,7
95  BITBLT Masque TO Xo%,Yo%+1,L%,H%,7
96  BITBLT Masque TO Xo%-1,Yo%+1,L%,H%,7
97  BITBLT Xo%,Yo%,L%,H% TO Masque
98  XBIOS (,5,L Phys,L Phys,-1)
99  IPL 3
100 RETURN
101 '
```

Ligne 0-6: Préparation de l'écran selon la résolution. La variable «Np%» est le nombre de plans selon la résolution.



Ligne 8: «Phys» est l'adresse physique de l'écran. «Ecran» est l'adresse de départ d'une réserve de mémoire divisible par 256, afin de s'en servir comme écran secondaire.

Ligne 10-13: On dessine quelques figures géométriques.

Ligne 15-32: Partie gérant la découpe d'un bloc à la souris.

Ligne 34: Voici une astuce pour calculer d'une manière sûre la taille en octet du bloc accueillant le BITBLT quelque soit la résolution. Ce calcul tient compte du nombre de plans (Np%).

Ligne 35-36: On réserve trois tampons mémoire : «Dessin» pour le bloc d'origine, «Masque» pour le bloc-masque, et «Buffer» pour mémoriser le fond sur lequel vont se déplacer nos blocs.

Ligne 38-41: Proposition de type de masque: plein ou évidé (in English to prepare Europe...)

Ligne 43-58: Au retour de la procédure choisie, le masque adéquat est dans le bloc-masque. On affiche notre bloc en deux fois: d'abord le masque avec le mode 4 (NOT Bloc AND Ecran), puis le dessin en transparence par dessus avec le mode 7 (OR). La condition de sortie de la boucle est la réponse «End» à la boîte d'alerte. Remarquez également la libération des tampons mémoire à la ligne 57, afin de permettre d'autres essais (donc de nouveaux calculs de tailles).

Ligne 59: Si le programme est compilé, on sort vers le bureau, sinon on retourne à l'éditeur.

Ligne 63: Procédure de calcul d'un masque plein.

Ligne 64: On interdit les interruptions pour gagner de la vitesse.

Ligne 65: On redirige les instructions graphiques vers un nouvel écran logique (écran que l'on s'est réservé en ligne 8), tout en gardant le même écran physique. Conséquence: le dessin du masque ne se voit pas à l'écran, puisque tout ce qui suit ce passe sur l'écran logique.

Ligne 66: On affiche le bloc d'origine (Etape 1).

Ligne 67: On dessine une boîte autour du bloc. On la dessine avec une marge de deux pixels de chaque côté afin «d'ouvrir le bloc» et d'être certain de remplir la boîte tout autour du bloc découpé (Etape 2).

Ligne 68: On remplit la boîte. Le motif de remplissage entoure le bloc découpé (Etape 3).

Ligne 69: On recopie le dessin d'origine sur le bloc obtenu en effectuant un (NOT Bloc AND Ecran) afin d'obtenir une ombre chinoise inversée (Etape 4).

Ligne 70: On recopie le dessin d'origine sur le bloc obtenu en effectuant un (NOT Ecran) afin d'obtenir le masque définitif (Etape 5).

Ligne 71: On mémorise le masque.

Ligne 72: On rétablit le niveau normal d'interruption.

Ligne 73: On rétablit l'écran logique sur l'écran physique.

Ligne 78: Procédure de calcul d'un masque détouré.

Ligne 79: On calcule d'abord un masque plein.

Ligne 81-88: On décale de deux pixels (cette valeur est arbitraire) dans les huit directions le masque, afin d'obtenir un masque plus grand que le dessin lui-même.

Ligne 89: On mémorise le masque.

Nous en avons terminé avec les transferts de blocs. Bien sûr, il existe d'autres applications. Seule votre imagination en fixera les limites. La prochaine rubrique amorcera un tout autre domaine, puisqu'il s'agit de la gestion des fichiers.

Pierre-Jean Goulier ■

LES RASTERS
Peindre comme monsieur Vincent

L'arrivée de la nouvelle version de Neochrome Master est l'occasion d'approfondir nos connaissances des rasters. Apprenons à utiliser les palettes de couleur dans une image.

Ciel d'orage. Les corbeaux menaçants n'annoncent rien de bon. Champ de blé torturé par le vent. Une fois de plus (la dernière?) Vincent Van Gogh recherche son idéal, la perfection de sa création. Une fois encore une nouvelle version de *Neochrome Master voit le jour.*

Comme à l'accoutumé ses généreux auteurs la destine au domaine public. *Neochrome Master* est gratuit (ou presque), disponible sur le 3615 ATARI que l'homme à l'oreille coupé aurait sans doute utilisé pour écrire à son frère Théo si le minitel avait existé en 1890.

Saluons l'effort de ses développeurs. *Neochrome Master* est devenu au fil du temps l'un des meilleurs programmes de dessin sur STE. De nouvelles fonctions, très intéressantes, apparaissent avec la version 2.26, dernière en date. Sa sortie est aussi l'occasion de revenir sur l'emploi des palettes multiples, une technique pas toujours facile à maîtriser.

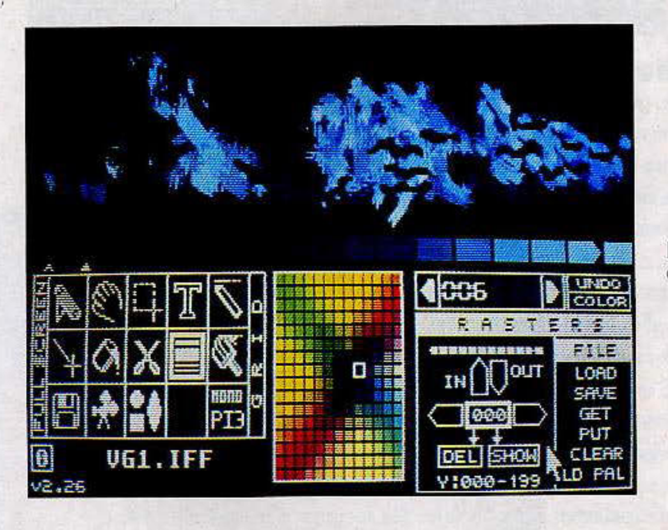


Figure 1: Peindre le ciel avec 16 nuances.

Grandeur et décadence

Il ne s'est écoulé que six mois depuis la disponibilité de la version 2.24, et déjà deux nouvelles versions ont pris le relais. Si la 2.25 n'a pas apporté grand chose de nouveau, en revanche, la 2.26 vient de s'enrichir d'une potentialité phénoménale: la prise en compte de modules autonomes.

Ces derniers sont en fait de petites routines qui vont permettre de rendre bien des services, et d'ajouter au gré des futurs créateurs des fonctions toutes plus inédites les unes que les autres.

Cette version est livrée avec les deux premiers modules disponibles: *Pi3conv* et *Loadspec*.

Le premier sert, comme son nom l'indique, à récupérer des images monochromes au format *Degas*. La conversion peut s'effectuer soit en 2 couleurs, soit en 4 niveaux de gris, cette dernière méthode se révèle nettement meilleure. Voici donc une extension intéressante et qui fonctionne bien.

On ne peut malheureusement pas en dire autant de *Loadspec* qui semble fortement buggé pour sa première apparition. Il nous a été, pour l'instant, impossible de le faire fonctionner sur un STE. C'est vraiment dommage, car l'idée est excellente: il s'agit de récupérer des images en provenance de *Spectrum*, le logiciel aux 512 couleurs. Attendons un peu et gageons que ce problème sera rapidement résolu.

Dans l'écran de présentation de *Neochrome*, deux cases ont été libérées pour accueillir ces nouveaux modules, ceci au détriment de la gomme et de l'airbrush (Nozzle). La gomme doit donc dorénavant être appelée par la touche [E] (Eraser) alors que le Nozzle est accessible par un clic droit sur l'icône des brosses. Les modules indépendants peuvent être chargés depuis le menu «Fill» sous les noms de User1 et User2.

LES OPTIONS DU MENU RASTERS

Mémoriser une palette
Récupérer une palette
Fleches pour le choix des rasters
Numéro du raster courant
Détruire le raster courant
Coordonnées verticales du raster courant
Visualiser la position du raster courant
RASTERS
FILE
IN
OUT
LOAD — Charger des rasters
SAVE — Sauver des rasters
GET — Mémoriser des rasters
PUT — Récupérer des rasters
CLEAR — Détruire tous les rasters
LD PAL — Charger une palette sur un ensemble de rasters
DEL
SHOW
Y:000-199

Le panneau des options Raster.

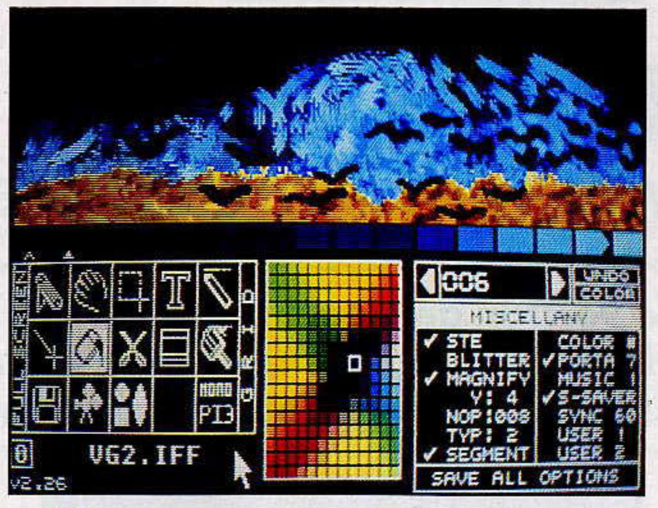


Figure 2: Utiliser des rasters intermédiaires pour éviter les horizontales trop franches.

Et quelques broutilles

A part cette nouveautée majeure, quelques changements sont intervenus depuis la version 2.24. Ainsi, les buffers du Jack Knife et du Copybox sont désormais interchangeables avec la touche F10.

La fonction Remap, qui permet de réadapter les couleurs d'un morceau d'image découpé au préalable, s'enrichit de la combinaison [Shift] [F5]. Cette dernière se transforme la zone d'image sélectionnée avec les couleurs situées entre les marqueurs gauche et droite de la palette.

De plus, quelques raccourcis-clavier sont ajoutés: [CTRL-L] pour charger des images, [CTRL-S] pour les sauver, [CTRL-X] pour quitter.

La copybox gagne en précision avec la visualisation de sa largeur et de sa hauteur en données numériques. Le collage d'une découpe peut se faire soit au centre de l'écran avec F3, soit à la position précise de la découpe avec Shift-F3.

Quelques autres petites modifications sont disséminées ça et là, mais sans grande importance. Les plus passionnés d'entre vous les trouveront sans doute pratiques.

La Rasterpeinture

Depuis qu'il est «Master» *Neochrome* offre la superbe possibilité d'utiliser les rasters de couleur. Inutile de se voiler la face: il est beaucoup plus facile de proclamer haut et fort: «Avec *Neochrome* on peut faire des images en 3200 couleurs!» que de passer à l'acte, même avec seulement une cinquantaine de teintes.

En règle générale, la pratique de la technique des rasters reste assez délicate.

Silence total. Battements d'ailes, croassements. Vincent s'accroche à son chevalet, sa tête est déjà ailleurs, là-haut. Et vous, néophyte du nouveau *Neochrome*, vous vous battez avec vos palettes. Rien ne va, la couleur refuse l'harmonie et vice-versa, les rasters s'entrechoquent. Pour vous guider dans la maîtrise de cette «Rasterpeinture», nous vous proposons quelques bons principes à travers deux exemples simples.

Palettes et rasters

La création des rasters est facile. Chaque ligne de l'écran peut comporter 16 couleurs. Pour les formats d'images simples, cela suffit. Cependant, l'écran basse résolution couleur contient 200 lignes et rien n'empêche le programme de changer de palette à chaque ligne, s'il est capable d'effectuer les changements assez vite pour afficher correctement l'image. *Neochrome* fait partie des (rares) logiciels qui relèvent ce défi. En théorie, le logiciel peut donc afficher 16x200=3200 couleurs simultanément. Cependant, il ne faut pas se leurrer, dans la pratique, on se contentera de beaucoup moins, parce qu'une image n'a pratiquement jamais une structure uniquement verticale.

Vous l'avez sans doute compris, le principal problème posé par les rasters est que, quoi qu'on fasse, on n'aura toujours que 16 couleurs par ligne. A partir de cette constatation, peindre avec beaucoup de couleurs devient un véritable casse-tête.

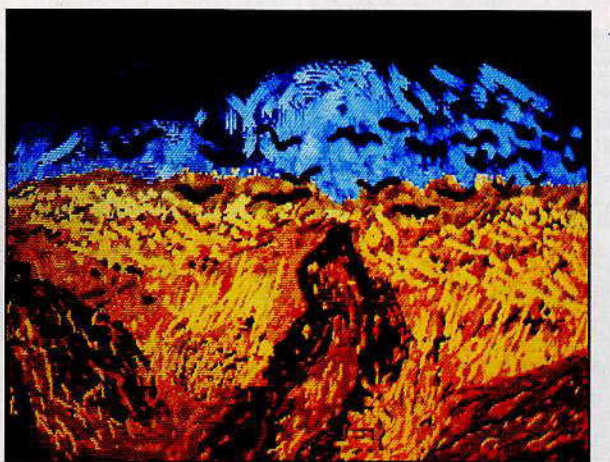


Figure 3: Le tableau est fini, merci aux rasters!.

Comprendre le menu

La première chose fondamentale à maîtriser est l'utilisation du menu Raster. Le premier raster (le numéro 0) commence toujours à la ligne 0. Si vous n'en avez défini aucun, il n'y a donc qu'une seule palette pour l'image jusqu'en ligne 200. Pour définir un raster, il faut régler le compteur-raster sur le numéro suivant (un clic sur la petite flèche droite) puis positionner le curseur dans l'image à l'endroit où vous souhaitez effectuer le changement de palette. Si vous ne faites rien d'autre, vous disposez bien de deux rasters mais toujours d'une seule palette. La suite logique consiste donc à modifier (intégralement ou partiellement) la palette de l'un des rasters, pour pouvoir disposer de plus de couleurs (jusqu'à 31 au maximum). 31 et non pas 32, car la couleur de fond ne saurait raisonnablement être changée sans générer d'étranges effets de contours d'images. Sur le même principe, on peut ensuite créer autant de rasters et de palettes que l'on veut.

L'option Save permet de sauver l'intégralité des rasters d'une image sur disque. Load sert à les recharger, rien de bien compliqué.



Figure 4: L'utilisation des rasters automatiques permet de simuler une infinité de fontes multicolores.

Put est l'équivalent de Save mais en mémoire vive, donc temporairement, et Get est le pendant de Load. L'opération Put/Get est très efficace lorsque vous désirez recopier des rasters sur un autre écran (pour faire de l'animation par exemple).

Au delà de la tourmente

Peindre à en perdre la raison. Mettre sa technique au service de la passion. Le fort contraste du ciel et du champ dévoile la hantise des extrêmes. La palette est durement malmenée. Le célèbre tableau de Van Gogh se prête très bien à une petite démonstration de l'utilisation des rasters. Le ciel, le champ, le chemin représentent trois niveaux bien distincts du tableau et sont, de surcroît, disposés horizontalement. Si l'on devait représenter cette ôuvre de l'artiste avec une seule palette de 16 couleurs, les choix à faire seraient terribles. De toute façon, l'image s'en trouverait appauvrie, le ciel comme le champ ayant besoin d'un maximum de nuances. La «rasterisation» s'impose donc.

Voici comment cette image a été construite (c'est un exemple raisonnable de ce qu'on peut envisager avec les rasters). La première partie réalisée est le ciel, peint avec une palette allant du noir d'encre à un bleu ciel assez clair. Cela constitue le premier raster qui s'étend jusqu'à la ligne 78 (voir fig. 1).

Ensuite, il faut commencer à peindre le champ. Cependant, pour éviter une coupure horizontale trop brute (principal écueil de la technique des rasters), il est fortement recommandé de prévoir un raster intermédiaire.

Ce dernier est doté d'une palette partagée en deux: une moitié pour les nuances du ciel, l'autre moitié pour le champ. Ce deuxième raster va des lignes 79 à 99. Il permet un passage souple entre deux zones bien différentes (voir fig. 2).

On pourra achever l'image par deux grands rasters. Le premier s'étend des lignes 100 à 150, pour les nuances du champ (jaune) et le haut du chemin (vert). Le second remplit le bas de l'écran, principalement avec des teintes rouge/brun et vert/brun.

Virtuoses de la palette

Pour faciliter le travail des palettes de couleurs sur les différents rasters, *Neochrome Master* met plusieurs outils à notre disposition. Dans le panneau des options de raster, la petite flèche vers le haut sert à mémoriser la palette courante. La flèche vers le bas permet de récupérer cette palette sur un autre raster. Ainsi, l'utilisation combinée de ces deux fonctions nous permet aisément de transférer les palettes d'un raster à l'autre. Pour cela, il ne faut pas oublier de changer de raster courant à l'aide des flèches horizontales. Autre précaution à prendre: pour être sûr de travailler avec la bonne palette de couleurs, il faut de temps à autre appuyer sur la barre d'espace, qui a pour effet de sélectionner la palette correspondant à la position courante de votre curseur. Une fois tous ces bons principes acquis et appliqués, il vous restera à sauver vos chefs-d'œuvre au format IFF, le seul capable de mémoriser la définition des rasters de couleur.

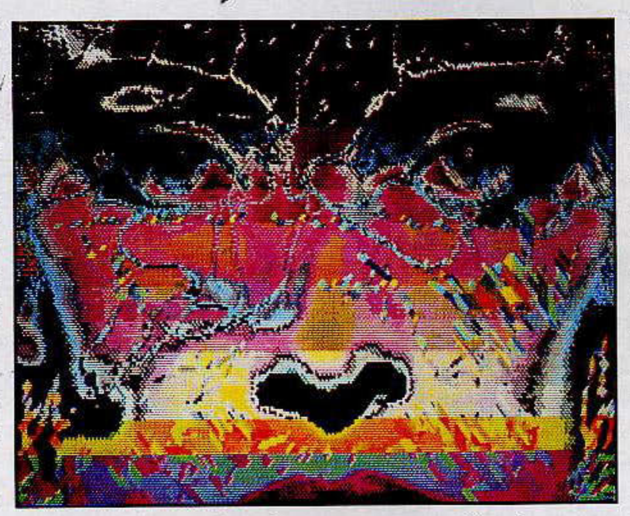


Figure 5: Bien utiliser les rasters pur créer des images spectaculaires.

Rasterisation automatique

Les plus perspicaces d'entre vous auront certainement remarqué qu'il reste une option dont nous n'avons pas encore parlé:LD PAL Nous l'avons gardée pour la fin, car il s'agit de la fonction la plus complexe et la plus puissante du menu Raster.

Son but est d'affecter une palette de couleur entière sur un même registre de couleur, mais répartie sur plusieurs rasters.

Pour bien comprendre cette technique avancée, il faut suivre le processus logique. En premier lieu, il faut utiliser une possibilité supplémentaire: la création de rasters en série. Pour cela, il est nécessaire de déterminer normalement le premier raster, puis de faire avancer le compteur d'un certain nombre (10, 20, 30,... au choix). Il reste alors à positionner le dernier raster avec le clic droit de la souris et le tour est joué! Le logiciel calcule la position de tous les rasters intermédiaires. Par exemple, si le premier commence en ligne 0, que vous avancez le compteur de 10, et que vous cliquez à droite en ligne 200, vous obtiendrez 10 rasters de 20 lignes chacun.

LD PAL enrichit énormément cette possibilité. Imaginez que vous vouliez faire un titrage multicolore. Qu'il soit dessiné ou écrit avec la fonction texte, on va pouvoir le créer rapidement avec une seule couleur (la n°1 par exemple).

Pour entraîner à signer notre beau tableau, écrivons le mot «Van Gogh» avec une police de taille 28, avec la couleur n°1. Puis définissons 16 rasters, dont le premier commence juste en haut des lettres et le dernier se termine en bas (avec la méthode automatique). Supposons que nous ayons une palette (.PAL) arc-en-ciel sur notre disquette. Il suffit alors de placer le compteur de rasters sur le premier raster correspondant à notre titre et d'activer la couleur n°1. LD PAL nous permet de charger notre palette arc-en-ciel. Et là, ô miracle, «Van Gogh» prend de la couleur! L'explication est simple: chaque couleur de la palette chargée est appliquée au raster correspondant (la couleur 1 pour le raster 1, la 2 pour le raster 2, etc.)

Ainsi, dans l'exemple de la figure 4, le texte est écrit sur ... 75 rasters, pas un de moins! La réalisation totale de cet écran a pris au maximum trois minutes, le plus long ayant été d'écrire le texte.

Vincent n'avait qu'une palette

1890, Van Gogh s'éteint, son œuvre va bientôt naître au monde entier. Sa palette, son unique palette ne l'a pas sauvé de la folie autodestructrice. Triste fin.

1992, *Neochrome Master*, est l'un des outils favoris des infographistes de l'univers Atari. Son mode multipalette basé sur la technique des rasters le sauve de l'oubli et relance ce logiciel qui supporte la comparaison avec les meilleurs. S'il comporte encore quelques imperfections dans son interface, le travail acharné de ses auteurs lui promet un bel avenir.

Alain Lioret ■

PALETTES ETENDUES

Des couleurs comme vous en rêviez...

Vous êtes-vous déjà trouvé en présence d'un ami qui se vante de pouvoir afficher 4096 couleurs sur son ordinateur? Faites-lui la nique: annoncez fièrement que votre STE en affiche 29 791!

J'en vois déjà au fond de la salle qui se disent «Je le crois pas, c'est pas possible.». Et bien si, c'est possible et nous allons vous le prouver tout de suite. Vous aurez même droit à un programme d'exemple, qui vous permettra d'afficher les images de *Quick Ray Trace* et de *DKB Ray Tracer* (tout deux disponibles sur le 3615 ATARI) en 29791 nuances sur STE ou 3375 sur STF.



La transparence enfin réalisable sur STE.

La ruse...

Tout d'abord, une petite mise au point: il ne s'agit pas ici d'afficher 29791 couleurs en même temps à l'écran, mais d'étendre la palette de 4096 couleurs du STE à 29791 couleurs. Notons que si l'on affiche 29791 nuances sur un STE, un STF en affichera 3375, ce qui n'est déjà pas si mal.

Niveau écran 1	Niveau écran 2	Niveau obtenu	Soit niveau
0	0	0	0
0	1	0.5	1
1	1	1	2
1	2	1.5	3
2	2	2	4
2	3	2.5	5
3	3	3	6
3	4	3.5	7
4	4	4	8
4	5	4.5	9
5	5	5	10
5	6	5.5	11
6	6	6	12
6	7	6.5	13
7	7	7	14

Est-il besoin de rappeler qu'une couleur est formée par l'addition de trois couleurs de base (que l'on appellera composantes) qui sont le rouge, le vert et le bleu. Chaque composante ayant une intensité différente, cela permet d'obtenir toutes les nuances du spectre lumineux. Pour afficher une couleur, le STE a donc besoin qu'on lui fournisse le taux de ces trois composantes.

Sur STF, on a 8 niveaux par composante, ce qui fait 8x8x8=512 nuances alors qu'un STE dispose de 16 niveaux soit 16x16x16=4096 nuances.

On se rend compte que pour augmenter le nombre des couleurs, il faut accroître la quantité de niveaux de chaque composante.

Comment peut-on multiplier le nombre de ces niveaux? C'est très simple: si l'on affiche un point d'une certaine couleur, par exemple un bleu (0,0,9) et tout de suite après le même point mais d'un bleu un peu plus clair (0,0,10), notre œil verra le point de couleur (0,0,9.5).

C'est-à-dire, en fait, la moyenne des deux couleurs. Cela est dû à la persistance de l'œil. En effet le clignotement rapide de deux couleurs trompe notre œil qui voit la moyenne de ces deux couleurs.

Voilà qui est intéressant, il suffit donc d'afficher à tour de rôle deux couleurs légèrement différentes pour voir apparaître une couleur intermédiaire.

Voyons un peu ce que cela donne par niveau de rouge, vert, ou bleu pour un STF.

On obtient donc 15 niveaux au lieu de 8, soit 15x15x15=3375 couleurs. Pour les STE c'est la même chose, si ce n'est que l'on ob-

Tout ce qui scintille n'est pas or!

Le fait d'afficher deux images fait scintiller l'écran. Dans certains cas, en particulier sur STF, cela peut devenir vraiment gênant. Il y a une solution assez satisfaisante qui consiste à passer en 60 Hz. L'écran étant rafraîchit tous les 60ème de seconde, le scintillement est alors beaucoup moins net. Pour effectuer cette manœuvre, vous pouvez utiliser le petit programme HZ5060 disponible sur le 3615 ATARI.

tient 31 niveaux, au lieu de 16, soit 31x31x31=29791 nuances!

Vous allez dire «c'est de la triche, c'est pas vraiment des nouvelles couleurs», et bien attendez un peu de voir les images de *QRT* ou *DKB*!



La brillance au niveau True Color.

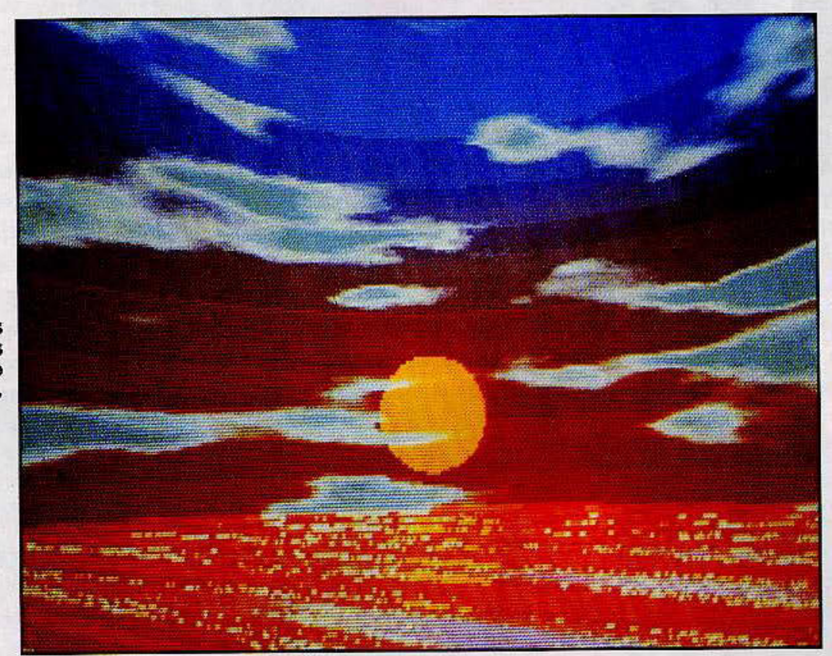
Travaux Pratiques

Comme promis, nous allons illustrer cette technique par un programme qui affiche les images RAW de *Quick Ray Trace* (.QRT) ainsi que les images TARGA24 (.TGA) de *DKB Ray Trace*.

Son fonctionnement est simple. Saisissez le programme *29791.S* (voir encadré «Disquette spéciale listings») avec *DevpacST* 2, assemblez puis exécutez-le (pour ceux qui ne disposent pas de cet assembleur ou qui ne désirent pas saisir le listing, la version du programme directement exécutable est disponible en téléchargement sur le 3615 ATARI). Le programme vous demande alors le nom d'une image provenant de *QRT* ou *DKB*. Entrez le nom complet sans oublier l'extension (.RAW ou .TGA). Après une trentaine de secondes, vos efforts seront récompensés par l'apparition d'une superbe image très colorée.

Attention, le programme ne gère que les images TARGA24 type 2 (99% des images TARGA24), les RAW provenant de *QRT* pour ST et les images RAW provenant de *QRT* pour Amiga. Toutefois l'extension de ces dernières devra être modifiée en .RAM, pour que le programme puisse faire la différence.

Pour voir une autre image, appuyez sur la barre d'espace. Pour sortir du programme tapez sur [CONTROL]+[C].



Des nuances de couleurs proches de réalité.

Happy end

L'utilisation de ce programme est très facile. Nous n'avons pas voulu, par souci de simplicité, entrer dans les détails techniques de sa programmation en assembleur.

Si toutefois vous avez des questions de cet ordre à poser, vous pouvez le faire sur le 3615 ATARI en bal Humfrey.

Mathias Agopian

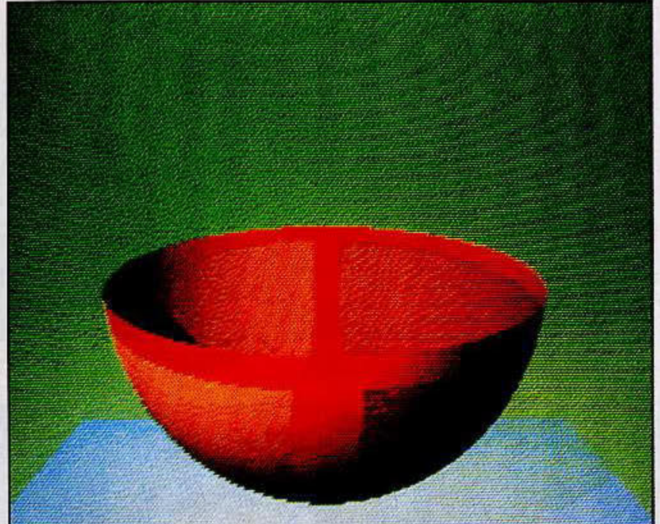
LE LISTING

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer les programmes correspondants.
Pour vous les procurer, téléchargez-les sur le 3615 ATARI ou procurez-vous la disquette correspondante (voir bon de commande encarté à la fin de ce numéro).

QUICK RAY TRACE (V)
Sculptures quadriques

De l'ellipsoïde au cylindre en passant par l'hyperboloïde, nous vous emmenons faire un petit tour au milieu de ces formes issues des mathématiques et portées à l'écran grâce au lancer de rayon.

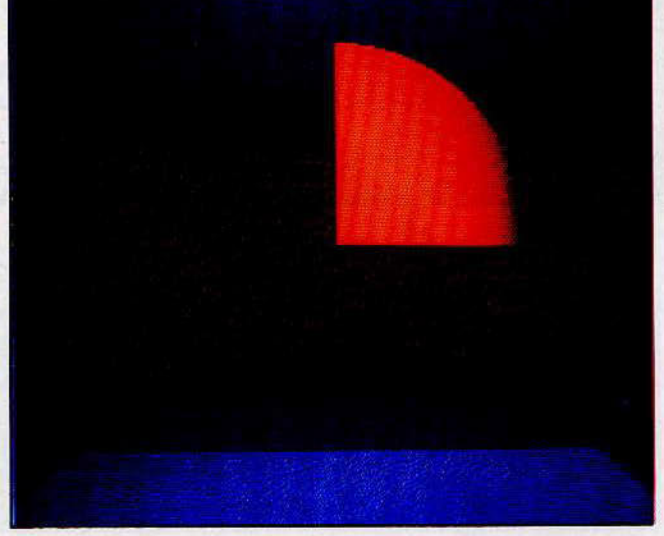
Nous avons bientôt fait le tour des possibilités du logiciel *Quick Ray Trace* (c.f. Atari Magazine n°30 à 33). Parce qu'elles sont un peu plus difficiles à mettre en œuvre, nous avions délibérément laissé de côté les surfaces quadriques (ou quadratiques).
Celles-ci complètent agréablement les primitives de base que nous avons découvert précédemment: sphères, plans, triangles, anneaux.



Une demi-sphère avec texture brique: le bol (listing n°2).

Pour un quartier d'orange

Pour bien utiliser la primitive `Quadratic`, il est fondamental de comprendre l'importance des paramètres à fournir. Ces derniers suivent des règles strictes qui répondent à des exigences mathématiques que l'on ne pourra pas contourner sous peine d'obtenir à peu près n'importe quoi. Ainsi, nos bons vieux manuels de maths, que nous nous sommes empressés de jeter dans les greniers, nous rappellent la définition d'une sphère: $x^2+y^2+z^2=R^2$ (où R est le rayon de la sphère).
Si on l'applique à la syntaxe d'une surface quadrique avec *QRT* qui est: $ax^2+by^2+cz^2=d$, cela implique que l'on doit avoir `a=b=c=1` si l'on souhaite générer une sphère. D peut être un nombre quelconque strictement positif et égal à R^2. Mais, direz-vous, à quoi peut bien servir de créer une sphère avec une surface quadrique alors qu'une primitive peut le faire? Tout simplement, à obtenir des sphères partielles. En effet, il existe six paramètres supplémentaires qui permettent de limiter la construction suivant les trois axes X,Y et Z. Ces valeurs sont des données relatives par rapport à la position de l'objet défini par la commande `LOC`.



Un huitième de sphère (listing n°1).

Par exemple, en supposant que $d=R^2=1$, on peut fixer `ymin=0` si on veut tracer uniquement la moitié haute de la sphère (dans ce cas, on peut donner: `xmax=1`, `xmin=-1`, `ymax=1`, `zmax=1`, `zmin=-1`).
A nous les parts d'orange, les bols et autres objets sphériques de toutes dimensions. Les listings n°1 à 3 qui accompagnent cet article donnent des exemples d'utilisation de sphères partielles.

Pastèques et ballons de rugby

Le deuxième type de surfaces quadriques est l'ellipsoïde. Elle diffère de la sphère par des rayons distincts selon les axes.
La définition d'une ellipsoïde est:
$x^2/a^2+y^2/b^2+z^2/c^2=1$.
Nous devons donc fixer d à 1, et avoir a,b et c strictement positifs. Ces trois valeurs étant des dénominateurs (en bas), on comprendra facilement que si a est grand, le rayon en x soit petit et vice-versa. Le listing n°4 est un exemple d'ellipsoïde complète (on aurait pu aussi en créer des partielles).



Un quart de sphère: la noix (listing n° 3).

Les ellipsoïdes permettront de modéliser toutes sortes de formes basées sur l'ovale (pourquoi pas une tête humaine?).

Il était une bobine...

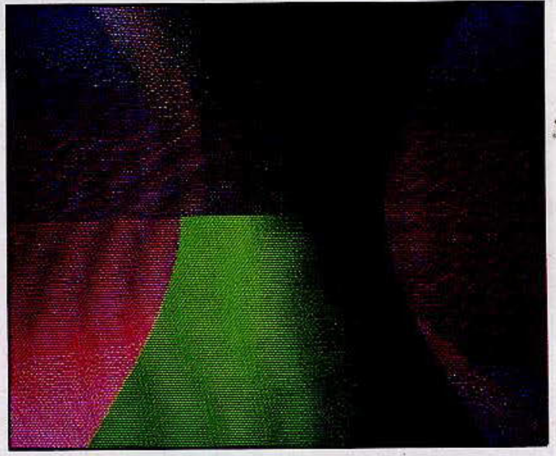
Les surfaces quadriques qui suivent sont un peu plus complexes et doivent, pour être valables, respecter des règles draconiennes.
Commençons par l'hyperboloïde à une nappe (et oui, c'est son nom officiel).
Son équation est de la forme:
$x^2/a^2+y^2/b^2-z^2/c^2=1$.
Le coefficient négatif n'est d'ailleurs pas forcément en Z, mais peut être en X ou Y. Les formes obtenues seront alors différemment orientées. Dans tous les cas, on doit avoir d=1, l'un des trois coefficients a, b ou c négatif et les deux autres positifs. Les résultats obtenus donnent des formes un peu comparables à une bobine de fil ou une poulie de treuil.

Les deux font la paire

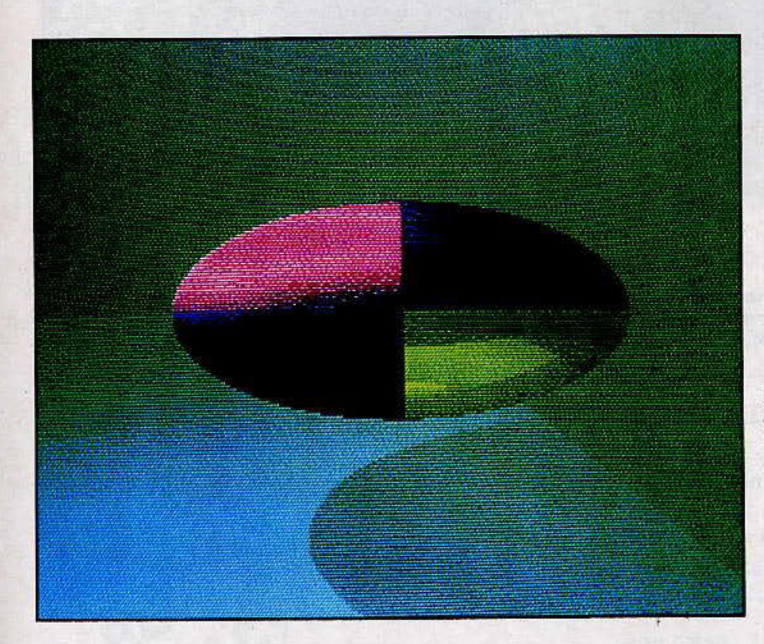
Autre surface quadrique du même genre: l'hyperboloïde à deux nappes (rassurez vous, c'est la dernière!). Dans ce cas, les mêmes principes que précédemment sont appliqués, mais deux coefficients sur trois doivent être négatifs. Le troisième doit obligatoirement être positif et d est toujours égal à 1.
L'équation est du type:
$z^2/c^2-x^2/a^2-y^2/b^2=1$.
Nous obtenons alors des formes composées de deux morceaux symétriques placés l'un en face de l'autre. Ce style d'objet, de par sa très grande particularité, est assez difficile à placer dans une scène tridimensionnelle. On ne voit pas bien ce que cela pourrait représenter, à moins de travailler dans l'abstrait. Le listing n°6 fournit un exemple avec b et c négatifs.

Le tube de QRT

Le cylindre est beaucoup plus intéressant, parce que bien plus fréquent dans le monde qui nous entoure. Il s'agit en fait d'un cylindre elliptique, car son rayon n'est pas forcément égal sur les trois axes. Il se construit à la condition que l'une des trois valeurs a, b ou c soit nulle. Les deux autres valeurs doivent obligatoirement être strictement positives, et d encore une fois égal à 1.
Le choix de la valeur nulle est fondamental pour déterminer l'orientation du cylindre. Si a=0 alors le cylindre sera horizontal, disposé suivant l'axe X, avec un rayon en Y proportionnel à b et un rayon en Z proportionnel à c. Si b est égal à zéro, le cylindre sera vertical et si c=0, il sera orienté en profondeur, face à nous. Tous ces positionnements restent valables uniquement si l'observateur est placé à l'origine et regarde droit devant. Pour d'autres choix, il faudra adapter ces résultats, mais si vous débutez, commencez par des choses simples. Dans le cas très particulier où les deux coefficients non nuls sont égaux, on obtient un cylindre circulaire de rayon R=a=b (si c=0 par exemple).
L'équation d'un cylindre est donc de la forme: $x^2/a^2+y^2/b^2=1$.



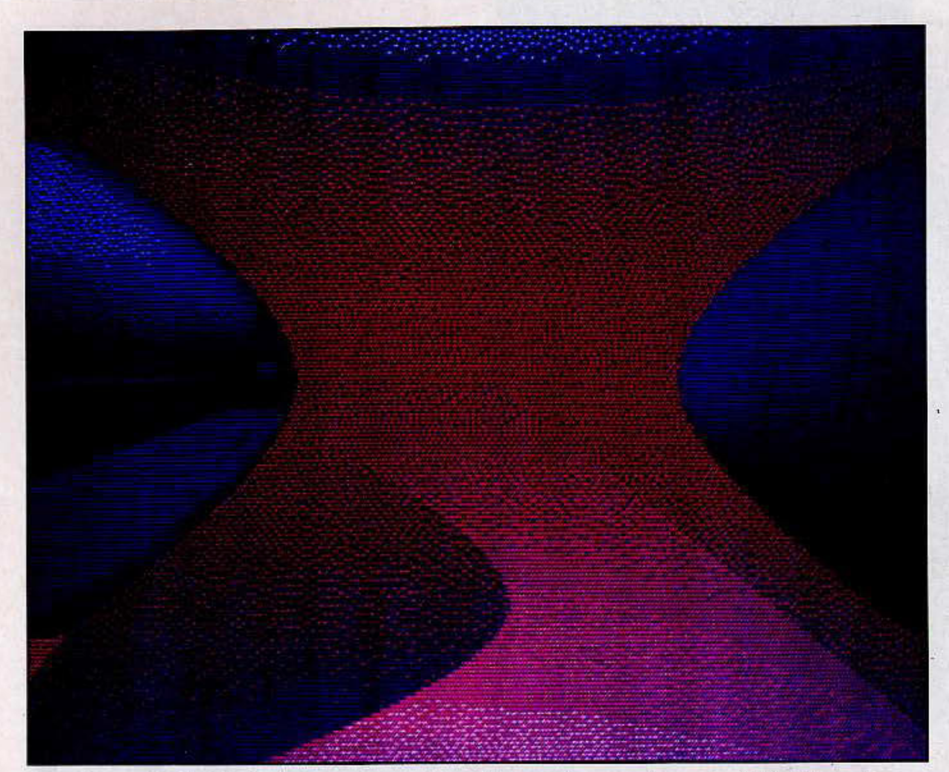
Une hyperboloide à une nappe à demi réfléchissante (listing n° 5).



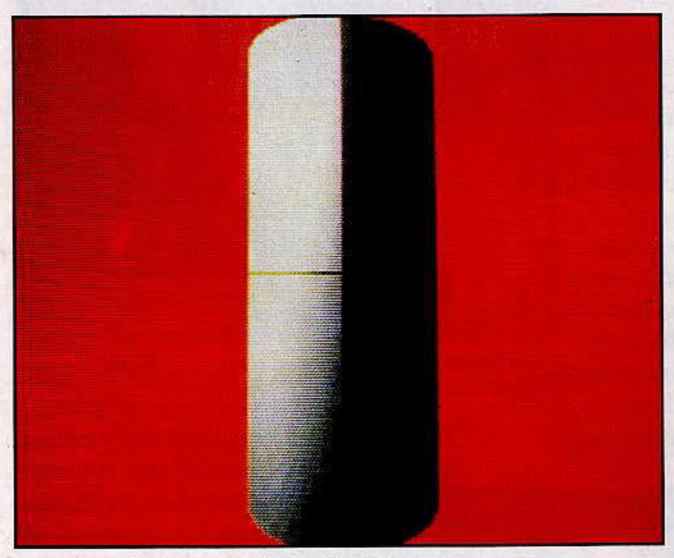
Une ellipsoide carrelée (listing n° 4).

Et le cône

Il reste encore un dernier type de surfaces quadriques: le cône elliptique. Plus encore



Une hyperboloide à deux nappes dont la partie gauche est réfléchissante (listing n°6).



Un cylindre elliptique (listing n°7).

que toutes les autres surfaces, il est important de respecter les contraintes de valeurs pour l'obtenir. On doit fixer d=0, et au moins l'une des trois autres valeurs à -1. Les deux valeurs restantes doivent être strictement positives. Le choix de la valeur égale à -1 est déterminant pour l'orientation du cône. L'équation d'un cône est de la forme: $x^2/a^2+y^2/b^2=z^2$.

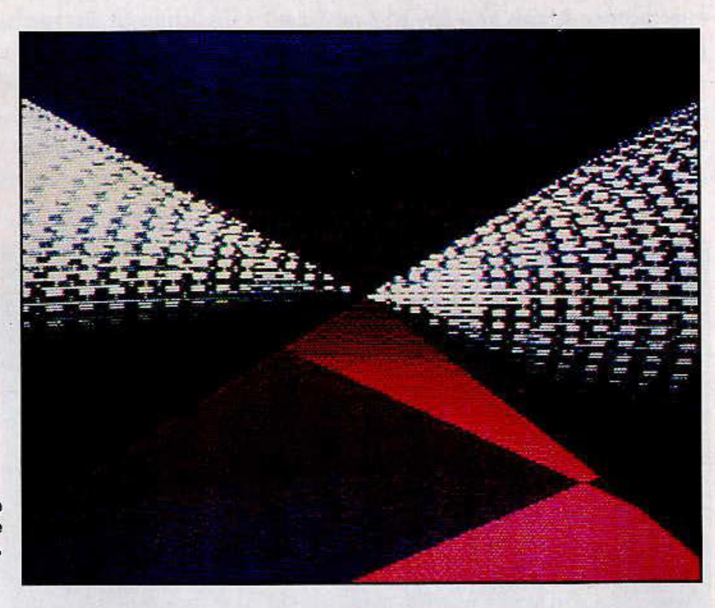
Comme pour le cylindre, le cône peut être circulaire si a=b. D'autre part, il faut signaler qu'il s'agit de produire un double cône symétrique. Si on ne veut garder que la moitié supérieure du cône, il faut fixer z>0 (avec zmin=0). Bien évidemment, ces résultats sont transposables pour x et y.

Nous avons fait le tour de toutes les surfaces quadriques réalisables avec *Quick Ray Trace*. Dans la réalité, il en existe d'autres, comme les paraboloïdes, mais celles-ci font intervenir des termes non élevés au carré, qui ne peuvent donc pas correspondre à l'équation type dont nous disposons.

Après ces quelques articles dédiés à ray-tracing, vous devez avoir en main la quasi-totalité des fonctions de ce merveilleux programme. Le mois prochain, nous apporterons les dernières touches pour conclure avec les possibilités de ce ray-tracer, avant d'en découvrir un autre (le fameux *DKB Tracer*) encore plus puissant.

En attendant, n'hésitez pas à réaliser de superbes images de lancer de rayon et envoyez-les au magazine: les plus belles seront sélectionnées et seront publiées dans notre grande saga dédiée au ray-tracing, le must de l'image de synthèse accessible sur STE.

Alain Lioret ■

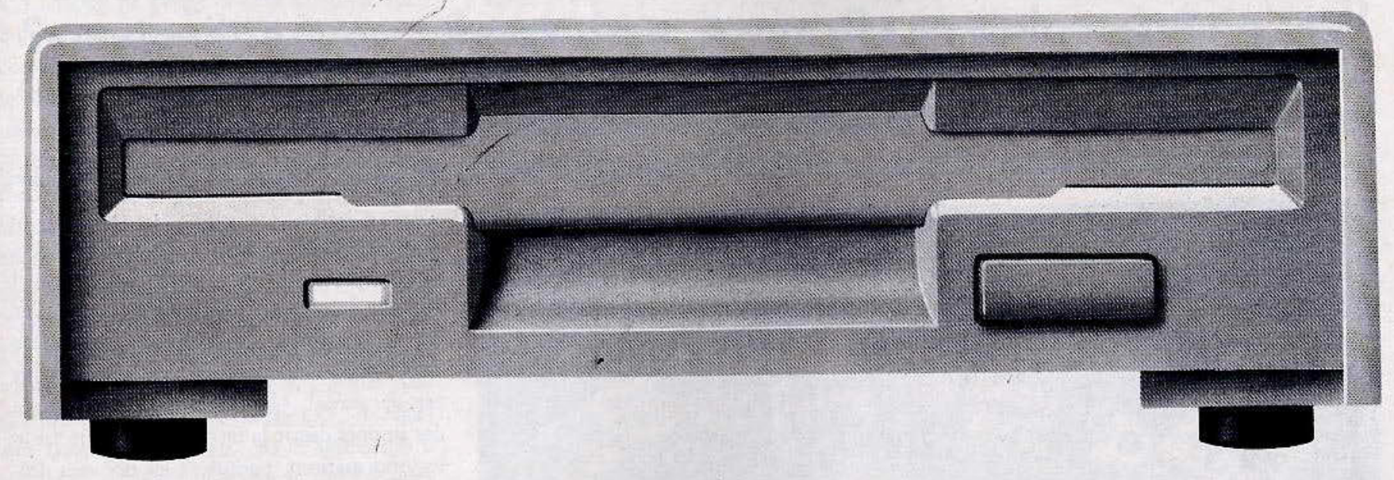


Un double cône avec texture (listing n°8).

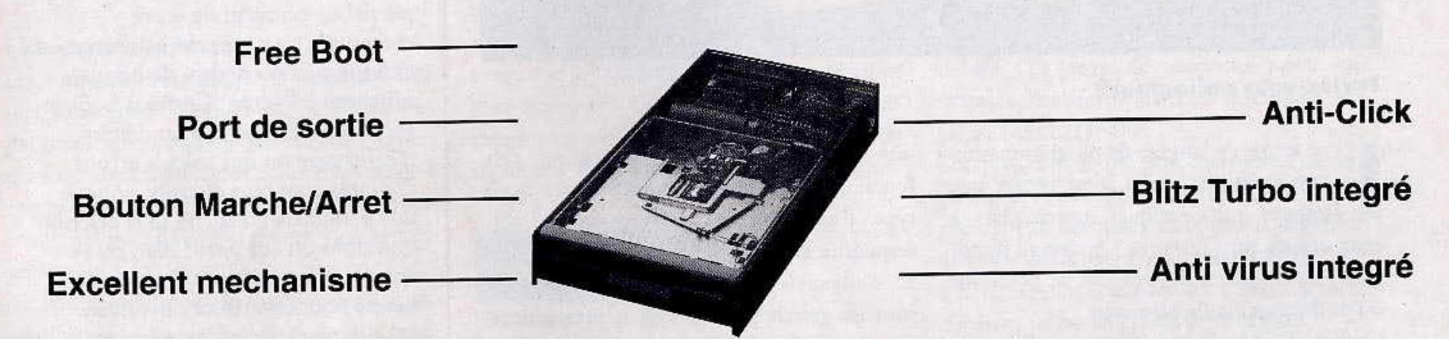
Le listing

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer les programmes correspondants.
Pour vous les procurer, téléchargez-les sur le 3615 ATARI ou procurez-vous la disquette correspondante (voir bon de commande encarté à la fin du numéro).

1,2,3,...
BRANCHEZ!!!



Virus, disquette endommagée, lecteurs bruyant,error disc... Y a t'il une fin au cauchemar de l'utilisateur Atari?
Chez **POWER**; nous voulons vous dire quoi brancher et avec quoi utiliser vos disquettes sans peur ni crainte des "**bad sectors**".En fait nous vous présentons le premier lecteur intelligent.Le **PC720B**.Inserez une disquette et commencez!!
Maintenant vous pouvez copier a toute vitesse, en utilisant le **blitz integré**, (que la disquette soit **AMIGA** ou **PC**).Si un virus est resident sur la disquette source le **PC720B** protegera le **boot block** de votre disquette destination et vous permettra de sauver vos fichiers.



Si ces ameliorations ne vous sont pas utiles; deconnectez les et utiliser le **PC720B** en lecteur silencieux , simple a utiliser.
Le **PC720B** est le seul lecteur integrant toutes ces possibilitées.Son prix est une deuxieme preuve de notre volonté de vous comprendre et de vous fournir le meilleur a chaque fois!!
Alors n'hesitez pas a dire aux autres personnes de le **BRANCHEZ!!!**

REVENDEURS CONTACTEZ NOUS

730 Frs

POWER COMPUTING FRANCE
15,Bld Voltaire 75011 PARIS
Tél: 43 57 01 69 Fax: 43 38 00 28

LES LANGAGES DE PROGRAMMATION
Dialoguez avec votre ordinateur

Vous parlez anglais ou allemand, mais «parlez-vous ordinateur»? Les langages de programmation vous ouvrent les portes des logiciels sur mesure et autres applications personnalisées.



Parlez-vous ordinateur?

Le terme de langage de programmation est trompeur, car il laisse penser que les langages informatiques ressemblent à ceux utilisés par l'homme. On devrait plutôt parler de systèmes de création de processus ou de description de processus.

Quoiqu'il en soit, connaître les différents «dialectes» que propose votre ordinateur pour communiquer avec le monde extérieur (vous!) peut vous aider à contenter quelques-uns des besoins que les logiciels sont inaptes à satisfaire (le besoin de création par exemple). Le célèbre proverbe ne dit-il pas «on est jamais mieux servi que par soi-même»?

Dans les domaines du ludique, du graphisme, de la musique ou de la gestion familiale les perspectives ouvertes aux programmeurs amateurs sont immenses. Les débuts sont parfois difficiles, semés d'embûches et de questions auxquelles on a du mal à répondre: qu'est-ce qu'un langage? Comment cela fonctionne-t-il? Lequel choisir? Pour quels types d'application? Quel est celui qui correspondra le mieux à ce que je désire faire? Et d'ailleurs que peut-on faire avec? Quels sont les grands principes de la programmation et les pièges à éviter?

Voici un guide à l'intention des Ataristes qui désirent aller un peu plus loin avec leur ST.

Qu'est-ce qu'un programme?

Un ordinateur est une machine très complexe qui sait exécuter un certain nombre de tâches simples, comme afficher des caractères à l'écran, faire de petits calculs, envoyer des caractères vers une imprimante, stocker des données sur disque, etc. En revanche, elle ne sait pas comment calculer vos impôts, jouer aux cartes et dessiner de belles images. Pour qu'un ordinateur exécute une tâche, il faut lui donner un programme. Ce dernier est la description complète de la manière dont l'ordinateur doit agir pour exécuter un certain travail. Un programme qui calcule des impôts décrit la façon dont il faut saisir les informations, comment les données doivent être stockées en mémoire, la manière dont les calculs vont être effectués, mais aussi comment les erreurs de fonctionnement

Les bogues

Un bogue est une erreur de fonctionnement qui peut entraîner un comportement imprévu d'un programme ou, carrément, un «plantage» de la machine (blocage partiel ou complet de votre ordinateur). Le type de «plantage» est indiqué par le nombre de bombes affichées à l'écran. Certains bogues sont évidents, comme un défaut d'affichage ou des calculs erronés. D'autres sont plus discrets, comme une erreur de calcul ne se produisant que dans un cas particulier, ou un «plantage» survenant de temps en temps pour une raison inconnue.
Les premiers ordinateurs étaient d'énormes machines utilisant des relais ou des lampes électroniques (comme les vieux téléviseurs).
Lorsqu'à la suite d'anomalies de fonctionnement les ingénieurs démontaient ces mastodontes, ils trouvaient le plus souvent des insectes. Ces petites bestioles étaient attirées par la chaleur que dégageaient les circuits électriques, et mangeaient l'isolant des câbles.
Bogue nous vient de l'anglais «bug» qui signifie insecte. Ce terme a été très vite adopté par le monde informatique.

vont être traitées, etc. Si la description est fausse, l'ordinateur fait automatiquement une erreur. Plus le programme accumule de tâches et plus sa description s'accroît.

Le langage de programmation

Un langage de programmation est un outil qui crée un programme, et qui définit le comportement de l'ordinateur. C'est un ensemble d'instructions qui gère plus ou moins facilement les fonctions de base de l'ordinateur. Par exemple, il existe des instructions d'affichage, de calcul, etc. Certains langages sont spécialisés et comportent surtout des instructions orientées vers un type de tâches spécifiques. Les exemples les plus classiques sont le *Cobol* qui facilite l'écriture de programmes de gestion, le *Fortran* les programmes scientifiques et le *Lisp* les programmes d'intelligence artificielle. Il existe des langages d'initiation pour apprendre à programmer très facilement et des langages non-spécialisés pour écrire n'importe quelle sorte de programme.

Les différents types d'instructions

Globalement, on rencontre quatre types d'instructions: les commentaires, les instructions de calcul, les instructions de contrôle et les instructions d'entrée/sortie.

Les commentaires

Les commentaires sont des instructions inutiles au programme, mais qui permettent au programmeur d'annoter son œuvre. Un programme commenté est toujours plus facile à comprendre et à modifier qu'un bloc d'instructions qui paraît, à première vue, incompréhensible. Par exemple, on peut commenter une routine dont le fonctionnement est relativement complexe. Grâce aux commentaires, on écrit directement dans le programme la date d'écriture, le nom du programme et quelques petites remarques personnelles.

Les instructions de calcul

Comme leur nom l'indique, les instructions de calcul effectuent des opérations mathématiques. Certains langages ne peuvent exécuter que les opérations mathématiques de base (addition, soustraction, multiplication, division), alors que d'autres possèdent un grand nombre d'opérateurs mathématiques. Tout dépend du type de langage et de sa vocation.

Les instructions de contrôle

Les instructions de contrôle sont très particulières, elles supervisent l'exécution du programme. Elles jouent le rôle soit de tests qui lancent l'exécution de tel ou tel bloc d'instruction (`si l'utilisateur a choisi cette option, faire ceci, sinon faire cela`), soit de séquences de répétition (`faire x fois telle opération`), ou encore de tests sur l'apparition ou la disparition d'un événement (`tant qu'il se passe ceci, alors faire cela`). Voici à quoi pourrait ressembler le module principal d'un programme de traitement de texte.

```
attendre frappe clavier
si caractère=code de commande
   si commande=DEL
      effacer dernier caractère
   si commande=flèche haut
      remonter position curseur
   si commande=flèche bas
      descendre position curseur
   si commande=flèche gauche
   déplacer curseur vers la gauche
   si commande=flèche droite
   déplacer curseur vers la droite
sinon ajouter caractère au texte
      afficher caractère
```

C'est une structure extrêmement simplifiée qui vous donne une idée sur la manière dont fonctionne un programme et sur l'énorme travail que représente l'analyse de tous les cas de figures qui se présentent lors de son exécution. Il faut que tout soit prévu et géré pour que le programme évite de se bloquer suite à une action imprévue. Notons que certaines personnes emploient le terme de structures de contrôle à la place d'instructions de contrôle. Les deux termes sont équivalents.

STPUZZLE

Apprendre en s'amusant!
A titre d'exemple, pour illustrer ce dossier, nous avons développé pour vous quatre versions d'un même jeu de puzzle (STPUZZLE) capable de traiter n'importe quelle image au format .PI1 ou .PI3. Vous disposerez ainsi d'un exemple de programmation en quatre langages différents (GFA Basic, STOS, C et Assembleur) afin de pouvoir comparer leurs spécificités et leurs performances respectives. Vous pourrez étudier leurs listings et vous détendre ensuite avec leurs versions exécutables.

Les instructions d'entrée/sortie

Les instructions d'entrée/sortie sont indispensables, car sans elles un programme n'a plus les moyens de communiquer avec l'extérieur. Les instructions d'entrée lisent des informations en provenance du clavier, de la souris et de la manette de jeu. Les instructions de sortie affichent des informations sur l'écran, envoient des caractères vers l'imprimante et émettent des signaux sonores. Il existe également des instructions qui lisent ou écrivent des informations sur une disquette ou un disque dur.

Un programme normal doit traiter un grand nombre d'informations. La nature de ces dernières dépend du type de programme: des nombres pour un programme de comptabilité, des noms et des textes pour un système de gestion d'une bibliothèque, des adresses et des numéros de téléphone pour un agenda, etc. Les programmes stockent toutes ces informations dans des variables.

Les variables sont des zones de mémoire identifiées par des noms. Pour accéder à une information, il suffit de donner le nom de la variable correspondante. Par exemple, un taux de TVA peut être stocké dans la variable t, la variable `tva` ou, carrément, la variable `taux_de_tva`. La plupart des langages réalisent facilement des calculs entre variables. La formule suivante peut, par exemple, être écrite dans de nombreux langages: `résultat=a+b`

Un programme un peu complexe contient parfois des centaines, voire des milliers de variables. Ces dernières comportent des informations de différentes natures. Il existe donc des variables spécialisées qui traitent plus facilement tel ou tel type d'information.

Pourquoi ne pas avoir un seul type de variable? Tous simplement afin d'accélérer la vitesse des calculs. Pour le microprocesseur du ST, il est beaucoup plus facile d'additionner deux octets que deux nombres de grandes dimensions. Plus la variable est longue et plus les opérations prennent de temps. De plus, il faut pouvoir stocker des informations non-numériques comme un nom ou un texte à l'aide de variables alphanumériques.

L'une des tâches importantes du programmeur est d'analyser toutes les informations

dont son programme aura besoin, et de prévoir des variables en conséquence. Au cas où ces dernières seraient trop nombreuses, il pourra recourir à des tableaux de variables.

Structure élémentaire d'un programme

En simplifiant, on peut considérer qu'un programme lit, avec les instructions d'entrée, des informations à partir de l'extérieur, effectue des calculs avec les instructions de calculs, et communique les résultats à l'utilisateur avec les instructions de sortie. Les premiers programmes informatiques fonctionnaient de la sorte. Les programmes actuels ont une structure beaucoup plus complexe, car ils sont interactifs et communiquent en permanence avec l'utilisateur. Ce dernier peut donner des ordres et en modifier les données à tout moment.

Environnement de développement

Un langage de programmation est essentiellement une série d'instructions spécialisées. Pour l'utiliser, il faut un environnement de développement, c'est-à-dire un ensemble d'outils qui permettent d'écrire un programme, de le tester, de le modifier et d'en sortir une version définitive. La plupart des environnements de développements actuels se présentent sous la forme d'un éditeur de texte éditant le programme, et d'une barre de menus déroulants grâce à laquelle on accède aux autres fonctions du système de développement.

Modes d'exécution des programmes

L'exécution des programmes par le microprocesseur s'effectue de deux manières différentes: compilée ou interprétée. Si la plupart des langages sont exécutés uniquement d'une manière ou d'une autre, certains, comme le *GFA Basic*, laissent le choix du mode d'exécution, combinant ainsi les avantages des deux méthodes.

Les langages interprétés

Le microprocesseur possède un langage qui lui est propre: le langage machine. Il est complexe et difficile à apprendre. Pour exécuter un programme écrit dans un langage spécialisé (*Basic*, *Pascal*, *Forth*, etc.), on utilise un «microprocesseur virtuel» qui exécute directement les instructions du langage choisi. On appelle ça un interpréteur, puisque ce «microprocesseur virtuel» doit analyser, donc interpréter, les instructions du langage avant de les exécuter. Un programme écrit dans un langage interprété est moins rapide qu'un programme écrit en langage machine, l'interpréteur étant lui-même écrit en langage machine. Il y a une perte de temps durant la phase d'analyse et d'exécution des instructions. Il existe beaucoup de langages interprétés car les interpréteurs sont des programmes relativement faciles à écrire, du moins pour des spécialistes du langage machine.

Les langages compilés

Plutôt que d'interpréter les instructions d'un programme, il est possible de le convertir en langage machine. C'est ce que font les langages compilés. Ces derniers sont plus performants que les langages interprétés. Le système qui fabrique le programme en langage machine, à partir du texte frappé par le programmeur, se nomme le compilateur. On pourrait croire que cette technique élimine totalement le besoin de langage machine, mais il n'en est rien. Un programme généré par un compilateur est moins performant qu'un programme écrit directement par un programmeur. La compilation s'apparente à la traduction automatique d'une langue en une autre, et vous n'êtes pas sans ignorer que ce problème est loin d'être résolu. Il reste néanmoins beaucoup plus facile d'utiliser un langage compilé que d'apprendre à programmer en langage machine.

Il faut distinguer le «programme source» (qui contient le texte des instructions saisies par le programmeur) et le programme compilé (qui est produit par le compilateur). Le «programme source» peut aussi être appelé listing, source ou code source. Le programme compilé est nommé programme exécutable ou code exécutable.

Qu'est-ce qu'une routine?

Une routine est un ensemble d'instructions qui effectue une tâche bien précise. Ce peut être un ensemble qui réalise un tri dans une série de chiffres, ou un ensemble qui attend que l'utilisateur frappe sur les touches [O] ou [N] du clavier, ou encore un ensemble qui calcule quelle est la meilleure position d'affichage d'un message sur l'écran. Il s'agit en fait de «super-instructions» qui s'ajoutent aux instructions de base du langage. Les bons programmeurs écrivent des routines standard, indépendantes des programmes, réutilisables dans d'autres applications. Certaines sociétés commercialisent des bibliothèques de routines qui étendent les possibilités des langages de programmation. Ce sont souvent des bibliothèques gérant des problèmes de gestion ou des questions mathématiques.

Les différents types de langage

On peut grossièrement classer les langages de programmation ST en deux catégories: les langages d'initiation et les langages avancés. Dans ce dossier, vous trouverez plus de détails sur ces différents langages. Les langages d'initiation ST sont tous dérivés du *Basic* (*GFA Basic*, l'*Omikron* et le *STOS*). Ce sont des langages qui permettent de s'initier à la programmation, mais aussi de réaliser des programmes relativement performants. Les langages avancés les plus courants sur ST sont le langage C, le *Pascal*, l'*Assembleur*, le *Forth* et le *Lisp*. Il se révèlent à la fois puissants et complexes, et sont utilisés par les programmeurs professionnels pour écrire les logiciels du commerce. Le *Lisp* est un cas un peu particulier car il est conçus pour traiter des problèmes d'intelligence artificielle.

Ecrire un programme

L'écriture d'un programme s'opère normalement, en plusieurs phases: étude des fonctionnalités du programme, étude technique, écriture et tests des routines de base, écriture du programme proprement dit et mise au point.

Les fonctionnalités

La phase d'étude des fonctionnalités d'un programme s'effectue à l'aide de papier et d'un crayon. Il s'agit de définir toutes les fonctions, ainsi que les méthodes utilisées pour les gérer. Il n'est pas possible d'écrire

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer ces programmes. Pour vous les procurer (ainsi que tous les autres listings d'Atari Magazine n°34), téléchargez-les sur le 3615 ATARI ou procurez-vous la disquette Spéciale listings (voir bon de commande à la fin du numéro).

un programme génial sans avoir la moindre idée de la manière dont il effectuera ces traitements.

La Faisabilité

L'étude technique permet de s'assurer que le programme est réalisable, et les méthodes de traitement choisies utilisables sur ordinateur. En théorie, on pourrait réaliser une belle animation en affichant toute une série d'images les unes après les autres. En pratique c'est impossible, car une image prend 32 Ko. On ne peut donc stocker plus d'une dizaine d'images dans la mémoire d'un 520 ST. C'est l'exemple même d'une technique séduisante sur le papier et inutilisable en pratique.

Les routines

Ecrire les routines nécessaires au programme équivaut à créer de nouvelles instructions. Ces routines doivent être bien testées car il est, par la suite, toujours très difficile de trouver un bogue situé dans une des routines de base du programme.

Programme et mise au point

L'écriture du programme doit s'effectuer assez rapidement puisqu'il suffit d'assembler des routines déjà créées. En revanche, la phase de mise au point est plus longue car il est nécessaire de rechercher les bogues (voir encadré). La meilleure manière de savoir si un programme est bogué, est de le mettre entre les mains d'une tierce personne.

Importance de l'interface utilisateur

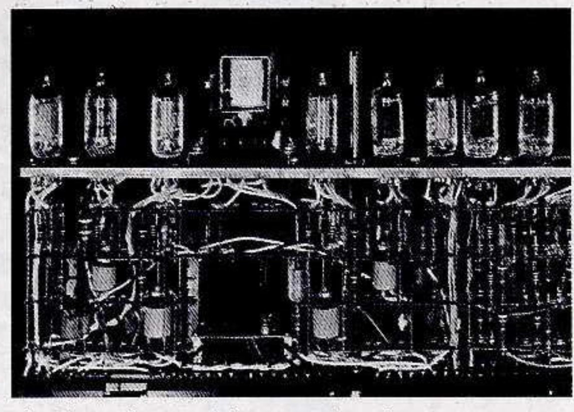
L'interface utilisateur est l'ensemble des moyens par lequel le programme et l'utilisateur communiquent. Cette interface doit être simple d'utilisation. L'ergonomie est la science des rapports entre l'homme et la machine, c'est pourquoi on dit qu'un programme simple à utiliser est ergonomique ou possède une bonne ergonomie, et inversement.
De base, le ST possède une bonne ergonomie avec la souris, l'écran graphique et les menus déroulants. Les concepteurs de logiciels devraient passer autant de temps à concevoir leurs interfaces utilisateur plutôt que d'écrire des programmes très performants. Un programme très puissant et peu pratique n'est pas intéressant car l'utilisateur passe trop de temps à comprendre la manière dont il fonctionne et fait souvent des erreurs de manipulation.

Patrick Leclercq ■

PATIENCE ET LONGUEUR DE TEMPS...
Petite histoire des langages

Afin d'apprécier la souplesse d'utilisation et la puissance des langages actuels de programmation, il est intéressant de savoir comment l'on programmait les premiers ordinateurs.

Plusieurs ordinateurs primitifs existaient un peu avant la Seconde Guerre Mondiale. Ils étaient utilisés par les militaires pour faire des calculs scientifiques et du décryptage de codes secrets. Ce n'était pas vraiment des ordinateurs, mais plutôt de grosses machines à calculer. Pour changer la programmation, il fallait ouvrir les machines et modifier les câblages. Bref, c'était vraiment la préhistoire!



Circuits électroniques d'un ordinateur de la première génération.

Le langage machine

Un peu après la Seconde Guerre Mondiale les premiers ordinateurs firent leur apparition. Ils coûtaient extrêmement chers et se programmaient uniquement en langage machine. Les programmeurs entraient les codes numériques des instructions avec de petits claviers numériques. Heureusement, les cartes perforées simplifièrent rapidement la saisie des données. Les programmeurs écrivaient leurs programmes à la main, perforaient leurs petites cartes et faisaient lire le programme par un lecteur de carte. Vive le progrès!

L'assembleur

Un jour, un programmeur fatigué d'écrire les programmes sous forme de codes numériques eu l'idée de faire exécuter cette tâche par l'ordinateur lui-même. Il conçut alors le premier *Assembleur* en langage machine. Le fait d'utiliser des instructions alphanumériques et non plus des codes numériques représentait une immense évolution, surtout pour les nerfs des programmeurs. A cet époque le coût du temps machine était très élevé. Le meilleur programmeur n'était pas celui qui programmait le plus vite, ou le plus proprement, mais celui qui écrivait le code exécutable le plus compact et le plus rapide. C'était le temps de la bidouille géniale. De nos jours, seuls les auteurs de jeux vidéo et de démos graphiques gardent cet état d'esprit très particulier.



L'IBM 704, l'un des premiers ordinateurs ayant une mémoire à tores de ferrite (1954). Ses capacités, extraordinaires pour l'époque, permirent le développement du langage Fortran. Plutôt encombrant!

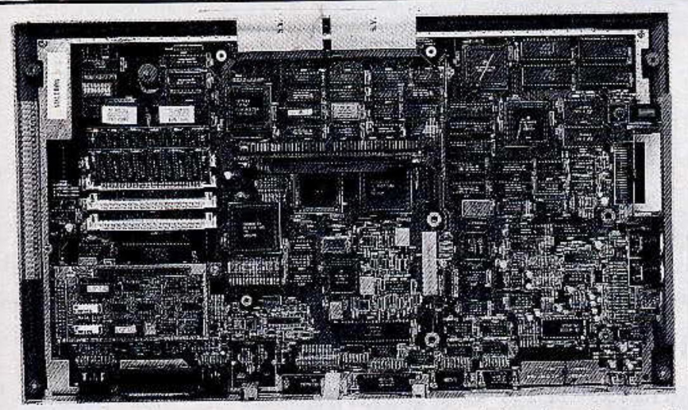
Les premiers langages évolués

Les premiers langages évolués dignes de ce nom furent le *Fortran* et le *Cobol*. Le *Fortran* est un langage scientifique qui fit son apparition en 1956. Afin de suivre les évolutions technologiques, il subît de nombreuses modifications. Le *Basic* était à l'origine une version très simplifiée du *Fortran*.

Le *Cobol*, quant à lui, est un langage conçu pour écrire des programmes de gestion. Il apparaît en 1960. Son auteur, une femme amirale dans l'armée américaine, est décédée il y a quelques mois. Ces langages sont tous compilés. Les programmes devaient être saisis par les programmeurs avec un clavier télétype et compilés la nuit, le temps machine coûtant plus cher le jour. Au petit matin, les programmeurs recevaient un listing im-

Le temps machine

Le temps machine est le temps d'exécution d'un programme. Plus un programme s'exécute lentement, plus il prend de temps machine. Le coût du temps machine d'un ordinateur peut être calculé en tenant compte du prix d'achat de la machine, de ses frais de maintenance et de sa durée de vie. Les premiers modèles d'ordinateurs avaient un coût d'utilisation très élevé. Les programmeurs devaient donc écrire des programmes utilisant le moins de temps machine possible.



primé contenant la liste des erreurs de compilation. Ils devaient trouver toutes les coquilles afin de recompiler le programme la nuit suivante! Ne croyez pas que ces techniques ont disparu. Elles sont encore pratiquées dans de grosses sociétés équipées de vieux systèmes informatiques. Etant donné que ces machines sont conçues pour travailler avec des cartes perforées, les données sont saisies avec de petits ordinateurs qui génèrent des fichiers simulant la structure de ces cartes. Ceux-ci sont ensuite envoyés dans la mémoire des gros ordinateurs.

Le début des temps modernes

Pendant que les industriels utilisaient le *Fortran* et le *Cobol*, les chercheurs en informatiques et les universitaires travaillaient. Pour preuve, il existe à l'heure actuelle plus d'un millier de langages. L'apparition des langages structurés, comme le *Pascal* et le langage *C*, fut une grande révolution. Simultanément, les premiers interpréteurs *Basic* permettaient à tout un chacun de s'initier aux joies de la programmation. La manipulation des programmes s'améliora également avec la généralisation des terminaux informatiques équipés d'écrans et de claviers.

Les temps modernes

Le grand changement des temps modernes, c'est l'apparition des ordinateurs individuels à coût réduit. Les programmeurs peuvent enfin s'amuser comme ils veulent sur leurs machines, au lieu de partager un gros ordinateur avec des dizaines, voire même des centaines d'autres utilisateurs. Les langages sont devenus interactifs, c'est-à-dire qu'il est possible de tester les programmes et de les corriger en même temps.

Les langages du futur

Pour l'instant, la future programmation semble s'orienter vers ce que l'on appelle les langages objets, c'est-à-dire des langages facilitant la création d'objets aux propriétés spécifiques et qui échangent des messages entre eux. Le *C++* est une adaptation langage objet du *C*. Il semble que ce soit le langage des années à venir. Affaire à suivre...

Patrick Leclercq ■

Les compilateurs

Le langage interne des ordinateurs est très complexe à utiliser. Afin de simplifier la programmation, les informaticiens ont conçu des langages de programmation plus simple: les langages évolués. Les compilateurs sont des outils informatiques qui prennent un programme écrit dans un langage évolué et le convertissent sous la forme d'un programme directement exécutable par l'ordinateur.

LES LANGAGES D'INITIATION
Etonnez-vous en Basic

La programmation d'un logiciel est une chose relativement complexe. Pourtant, les langages d'initiation permettent de s'adonner rapidement et simplement aux joies de la création informatique.

Les langages d'initiation ST les plus performants sont tous des dérivés du Basic. Ce dernier était à l'origine un langage simplifié provenant lui-même d'un langage scientifique: le *Fortran*. Il a vite connu un immense succès en raison de sa grande simplicité d'emploi et de sa facilité de programmation. Un bon programmeur en langage machine peut facilement écrire un *Basic* sur n'importe quel ordinateur. Le plus petit basic au monde est le *TBasic Portfolio* qui occupe moins de 2 Ko.

Tous les ordinateurs 8 bits des années 80 étaient équipés d'un interpréteur dont les capacités variaient selon le talent du concepteur. Les *Basic* les plus utilisés à l'heure actuelle sont le *GFA Basic*, l'*Omikron* et le *STOS*. Il existe également un *Basic* particulier dédié aux applications mathématiques: le *Basic 1000D*.

```
1000 screen copy 2,6*8,0,16*8,4*8 to physic,14*8,16*8 : extension #D 1 : wait 5
: screen copy 1,0,0,319,191 to physic,0,0 : return
1100 screen copy 2,0,0,6*8,6*8 to physic,10*8,8*8 : extension #D 2 : wait 5 : sc
reen copy 1,0,0,319,191 to physic,0,0 : return
1200 screen copy 2,16*8,0,24*8,8*8 to physic,22*8,6*8 : extension #D 3 : wait 5
: screen copy 1,0,0,319,191 to physic,0,0 : return
1300 screen copy 2,24*8,0,30*8,6*8 to physic,6*8,18*8 : screen copy 2,0,6*8,10*8
,10*8 to physic,2*8,65 : extension #D 5 : wait 5 : screen copy 1,0,0,319,191 to
physic,0,0 : return
1400 screen copy 2,10*8,6*8,12*8,8*8 to physic,18*8,4*8 : extension #D 7 : wait
5 : screen copy 1,0,0,319,191 to physic,0,0 : return
1500 screen copy 2,0,10*8,6*8,16*8 to physic,24*8,6*8 : extension #D 6 : wait 5
: screen copy 1,0,0,319,191 to physic,0,0 : return
1600 screen copy 2,6*8,10*8,12*8,16*8 to physic,28*8,14*8 : extension #D 8 : wai
t 5 : screen copy 1,0,0,319,191 to physic,0,0 : return

Banques mémoire réservées:
3  music   S:$1E6D00 E:$1E9300 L:$002600
4  3D      S:$1E9300 E:$1EA800 L:$001500
5  data    S:$1EA800 E:$1EFF00 L:$005700

Ok
```

Editeur du STOS. Il est nettement moins pratique à utiliser que les autres éditeurs Basic.

Qu'est-ce que le Basic?

Le *Basic* est un langage interprété, c'est-à-dire que les instructions des programmes *Basic* sont exécutées les unes après les autres par un programme que l'on appelle un interpréteur. Les interpréteurs sont des systèmes interactifs. Ainsi, il est possible d'exécuter une instruction ou un programme à n'importe quel moment. Par exemple, l'instruction PRINT affiche sur l'écran le résultat d'un calcul. En tapant directement PRINT 3*2, on obtient 6. C'est un exemple simpliste, qui montre bien le principe de l'exécution directe. Grâce à cette méthode, les débutants en informatique peuvent facilement étudier le fonctionnement des instructions. Pour exécuter un programme, il suffit d'utiliser la commande RUN. Si quelque chose ne fonctionne pas dans le programme, l'interpréteur s'arrête sur l'erreur et affiche un message décrivant ce qui s'est produit. Le programmeur peut alors corriger directement son erreur et relancer l'exécution du programme.

```
 FILE  FIND  BLOCK  MODE  GO  RUN    Y:    0  X:  0  SIZE:   4840  SPRDEMO.BAS

XBIOS (Rez,4)
PALETTE ,$135

IF Rez=0 THEN Sprdef=2:W8=320:H8=200:W16=16
IF Rez=1 THEN Sprdef=4:W8=640:H8=200:W16=32
IF Rez=2 THEN Sprdef=5:W8=640:H8=400:W16=16

PRINT "{f";
DIM Bomb(7,1),Sprback(8),X(8),Y(8),Dx(8),Dy!(8),Flag(8),Count(8)
FOR I=1 TO 8
   Sprback(I)= MEMORY(256)
NEXT
Buf= MEMORY(128)' Buffer for calculation
FOR I=0 TO 7
   Bomb(I,0)= MEMORY(128)
   Bomb(I,1)= MEMORY(128)
NEXT
Sread(Bomb(0,0))
Sread(Bomb(0,1))
Sread(Bomb(1,0))
Sread(Bomb(1,1))
```

L'Omikron possède deux éditeurs de programmes. L'un d'eux travaille avec des numéros de lignes et l'autre sans. Cette image vous montre un exemple de programme source sans numéro de lignes. Cet éditeur est moins facile à utiliser que celui du GFA Basic.

Les instructions Basic

Il existe deux niveaux d'instructions *Basic*: des instructions de base qui permettent d'écrire de petits programmes et des instructions plus complexes pour écrire des programmes performants.

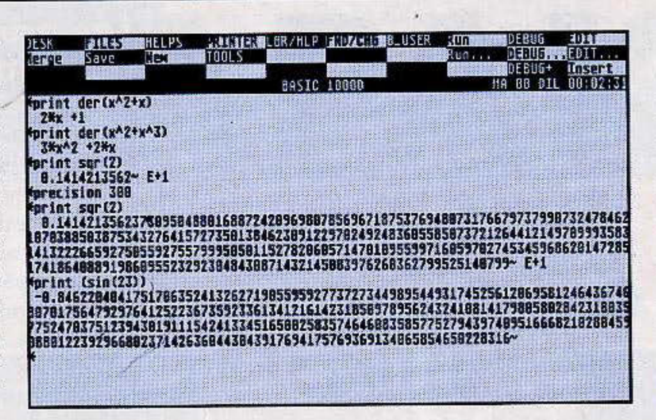
Les premiers *Basic*, uniquement conçus pour l'initiation, ne comportaient que les instructions de base. Les instructions supplémentaires ont été ajoutées par les constructeurs, afin d'augmenter la puissance de leurs langages et pour permettre une bonne gestion des capacités des machines (graphisme, son, disque, etc.)

Les compilateurs Basic

Même si les programmes compilés sont toujours plus performants que les programmes interprétés, les interpréteurs facilitent la mise au point des logiciels.

Actuellement, tous les *Basic* sont dotés d'un interpréteur et d'un compilateur.

Grâce à l'interpréteur on peut écrire et tester les programmes. Par l'intermédiaire du compilateur, on obtient une version performante et diffusable du logiciel, puisque celui-ci génère un programme directement exécutable par le ST.



Editeur du Basic mathématique 1000D. La valeur de la racine carrée de 2 est calculée entre 10 et 300 chiffres significatifs. L'instruction précision permet de choisir jusqu'à 1000 chiffres significatifs. Comme vous pouvez le constater, l'instruction der permet de calculer la dérivée d'une équation quelconque.

de lignes. Il existe peu d'ouvrages sur l'*Omikron Basic*, à part un manuel de programmation «avancée» proposé par l'éditeur lui-même.

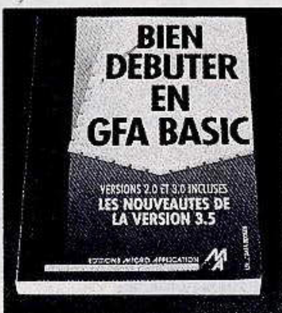
Par ailleurs, une bibliothèque d'instructions optionnelle facilite l'écriture de programmes GEM. C'est l'équivalent du *GFA Gup*.



Le STOS

Le *STOS Basic* n'est pas un langage structuré comme le *GFA Basic* ou l'*Omikron*. Il ressemble beaucoup aux anciens *Basic* des ordi







Le GFA Basic

Le *GFA Basic* est le langage le plus utilisé sur ST. Il a subi de nombreuses améliorations et en est actuellement à la version 3.5E (il existe une version 3.6 spécifique au TT). C'est un excellent langage intermédiaire entre le Basic traditionnel et les langages du type *C* ou *Pascal* (voir l'article sur les langages avancés). Sa vitesse d'exécution est excellente. Les programmes n'ont pas besoin de numéro de lignes et l'on peut créer des procédures à partir des instructions de base.

Les listings *GFA* ressemblent beaucoup à des programmes C ou *Pascal*. Les sources de documentation sur le *GFA*, aussi bien livres qu'articles, sont très nombreuses. Nous avons publié dans Atari Magazine (n°14-17-18-23 à 26) une série d'articles sur la programmation de jeux d'aventure et jeux de rôle en *GFA Basic*. Des produits complémentaires étendent les capacités de ce langage.

Par exemple, le *GFA Gup* est une bibliothèque de routines facilitant l'écriture des programmes GEM. Comme son nom l'indique, *GFA routines graphiques et sonores* contient de nombreuses routines supplémentaires. Hélas, ce produit est plutôt médiocre.

En revanche, le *GFA Assembleur* est un assembleur très performant utilisable seul ou en association avec le *GFA Basic*. De nombreux programmes ST ont été écrits en *GFA*, tant dans le domaine du jeu, que dans le domaine professionnel.

Par exemple, les jeux d'aventure *BAT* et *Iron Lord* sont à la fois écrits en *GFA Basic* et en *Assembleur*. De plus, il existe énormément de programmes en *GFA* dans le domaine public.

C'est un langage qui peut servir pratiquement à tout: initiation, graphisme, programmation de jeux d'aventure et de réflexion, gestion, serveurs minitels, etc.

Il existe même une version simplifiée du langage *Forth* écrite en *GFA Basic*.

L'Omikron Basic

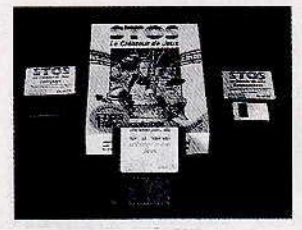
L'interpréteur et la documentation de l'*Omikron Basic* sont actuellement fournis gratuitement avec les ordinateurs Atari. Le compilateur doit être acheté à part.

C'est un langage performant tout en restant moins puissant que le *GFA Basic*. Sa grande force est d'être extrêmement compatible avec les *Basic* du type *Basic Microsoft* ou *GW-Basic*. De nombreuses personnes ont appris l'informatique avec ce type de *Basic* et apprécient de pouvoir exploiter leurs anciens programmes. Afin d'obtenir cette compatibilité tout en restant un langage moderne, l'éditeur de l'*Omikron* travaille de deux manières différentes: avec ou sans numéro

Le téléchargement

Le 3515 ATARI propose des listings Basic en téléchargement. Ces derniers ont été écrits par des programmeurs qui les ont versés dans le domaine public afin d'en faire profiter à un maximum de personnes.

Les fichiers qui suivent sont des listings du domaine public disponibles sur le 3516 ATARI. Le fichier NCC_1701.ARC contient des routines GFA Basic 2.0 et GFA Basic 3.0 capables d'afficher des images stockées dans les principaux formats graphiques du ST: Degas Elite, Degas Elite compacté, Neochrome, PaintWorks, Tiny et Spectrum 512. Le fichier SPRTGRBR.BAS est un programme en GFA Basic 2.0 permettant de créer des fichiers de sprites à partir d'images Degas Elite. Le fichier ROUTINES.LST contient des routines qui réutilisent, dans n'importe quel programme, les sprites de SPRTGRBR . Le fichier LOADEGAS.LST est une routine en GFA Basic permettant de charger une image Degas Elite sur l'écran. Le fichier SAVDEGAS.LST est une routine GFA Basic qui sauve le contenu de l'écran sous la forme d'une image Degas Elite. Le fichier PLYMUSIC.LST contient un programme jouant un air de musique. Le fichier VSUSRT.ARC contient des routines de tri écrites en GFA Basic. Le fichier OMIPLAY.AR contient le source d'un programme Omikron permettant d'écouter des sons digitalisés avec le logiciel ST REPLAY. Il est fourni avec plusieurs exemples de bruitages (mugissement de vache, cri d'éléphant, bruit de verre brisé, bruit de machine, etc.)





nateurs 8 bits, avec la numérotation des lignes et la possibilité d'y inclure un grand nombre d'instructions. Son intérêt provient de ses grandes capacités graphiques. La plupart des langages de programmation disposent de bibliothèques graphiques limitées, incapables, par exemple, d'afficher des sprites sans passer par des méthodes plus ou moins complexes. Le *STOS*, quant à lui, possède des instructions graphiques orientées vers la pro-

Editeur du GFA Basic. Les fonctions sont sélectionnées avec la souris ou les touches de fonctions. C'est un éditeur très souple et relativement performant.

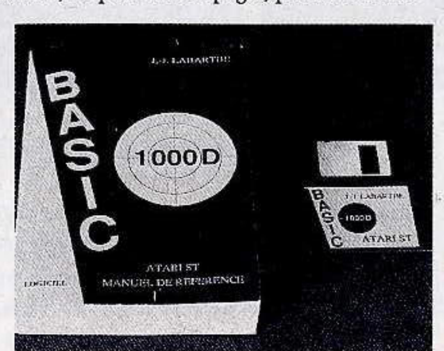
grammation de jeux. Il permet de créer des sprites animés qui se déplacent sur l'écran indépendamment du programme principal.

De nombreux produits complémentaires étendent les possibilités du *STOS* de base. Comme son nom l'indique, *STOS compiler* compile les programmes. *STOS Sprites 600* est une bibliothèque graphique contenant de nombreux sprites utilisables dans les programmes *STOS*. Si vous avez besoin d'explosions, d'arbres, de rochers, d'extra-terrestres, de vaisseaux spatiaux, de magiciens, de démons, d'animaux, etc., ne cherchez pas plus loin! *STOS Maestro* et *STOS Maestro Plus* permettent de créer des sons et des musiques en utilisant divers instruments ou des échantillons sonores digitalisés. *STOS Galore* contient des sources de jeux d'arcades écrits par des programmeurs Anglais. *STOS 3D* est un produit récent qui permet de créer et d'utiliser des objets 3D faces pleines. Il intègre un modeleur 3D pour créer ces objets 3D avec une incroyable facilité.

Le Basic 1000D

Ce *Basic* mathématique a été développé par un chercheur du CNRS. Ce logiciel n'est d'aucun intérêt pour un utilisateur ordinaire, mais fabuleux pour un étudiant ou un scientifique. Il est doté de capacités assez incroyables. La précision des calculs numériques réels ou complexes peut dépasser les 1000 chiffres significatifs. D'après la documentation, la racine carré de 2 est calculée avec 1000 chiffres significatifs en 1,3 seconde. Le plus impressionnant n'est pas la précision significative, ni le grand nombre de fonctions mathématiques, mais la possibilité de faire des calculs de manière symbolique.

Par exemple, le programme est capable de déterminer que la dérivé de a^2+b+5 est 2xa. C'est très impressionnant, sans même parler des opérations sur matrices, des recherches de racines, des développements de polynômes, etc. De plus, le compilateur permet encore d'accélérer la vitesse de calculs. L'épais manuel, de plus de 500 pages, présente toutes les possibilités de ce langage. Il existe une nouvelle version qui apporte quelques avantages mineurs sur STE, par ailleurs elle fonctionne sur TT.



Tout un univers à portée du clavier

En résumé, les outils ne manquent pas pour tous ceux qui désirent se lancer dans l'univers fabuleux, et au combien excitant, de la programmation. De la création d'un jeu à celle d'un utilitaire personnalisé (voir les quelques idées que nous vous soumettons en encadré dans ce dossier), vous ne regretterez pas les heures passées devant votre clavier.

Patrick Leclercq ■

Les structures de contrôle

Les structures de contrôle sont des instructions qui contrôlent le déroulement du programme. Les instructions de base à apprendre sont les instructions de tests (IF/ENDIF), les boucles répétitives (FOR/NEXT) et les boucles conditionnelles (DO/EXIT IF/LOOP). Les autres structures de contrôle ne servent que dans des cas particuliers, très rarement pour des programmes simples.

Les conseils du Dr Kendrix

Le graphisme

Le graphisme est un domaine qui fascine souvent les débutants. Comme nous l'avons vu plus haut, il ne faut pas essayer de comprendre tout de suite l'ensemble des instructions graphiques, mais se limiter aux instructions de base. Pour le GFA Basic, ces instructions sont COLOR (paramétrage de la couleur de tracé), LINE (affichage d'une ligne), PLOT (affichage d'un point), PBOX (affichage d'un rectangle de couleur), BOX (affichage des bords d'un rectangle), CIRCLE (affichage d'un cercle) et TEXT (affichage d'un texte en mode graphique). Vous pouvez réaliser de nombreux programmes graphiques avec ces quelques instructions.

Programmation modulaire

En règle générale, les premiers programmes sont une longue suite d'instructions incompréhensibles. C'est une erreur qu'il est préférable de ne pas commettre, et ce pour ne pas nuire à la lisibilité du programme, à la recherche des erreurs et à la possibilité de faire des modifications. Les langages structurés comme le GFA Basic et l'Omikron offrent la possibilité de définir des blocs d'instructions qui seront ensuite utilisés comme des super-instructions. On les nomme des procédures ou des routines. En GFA Basic, ces dernières sont créées par les instructions PROCEDURE et FUNCTION. S'ils sont bien conçus, les blocs d'instructions peuvent être réutilisés dans d'autres programmes.

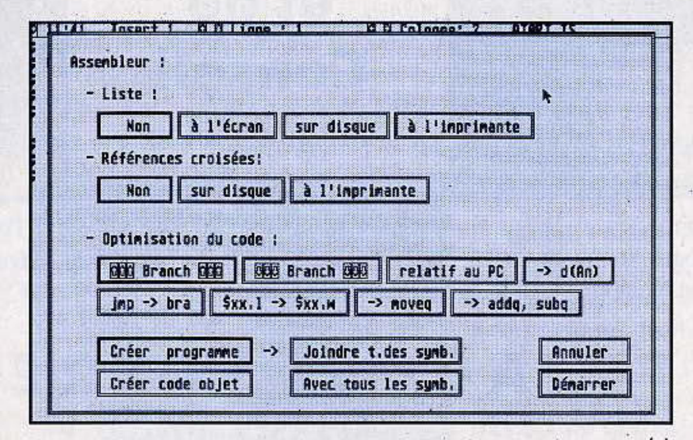
Présentation des programmes

Les programmeurs débutants sont contents lorsque les programmes fonctionnent. C'est bien, mais le côté fonctionnel n'est pas tout. Il faut que le programme soit simple à utiliser et que la présentation des données à l'écran soit claire.

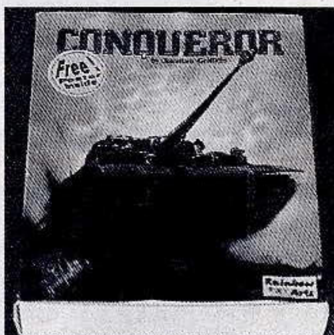
LES LANGAGES AVANCES
Connaître pour choisir

Les programmes de grande envergure ne peuvent être écrits qu'avec de puissants langages. Pour en choisir un, il faut avant tout bien connaître ses principales caractéristiques.

Après de nombreuses heures d'initiation à la programmation, quand votre trentième répertoire téléphonique «fait main» aura fini par vous lasser, peut être ressentirez-vous le besoin de passer à la vitesse supérieure. Il vous faudra alors adopter un de ces langages que l'on dit avancés, qui permettent de se lancer dans la programmation de logiciels beaucoup plus performants. Leur structure, le nombre de leurs instructions, ainsi que la vitesse d'exécution qu'ils autorisent, en font des outils, certes plus complexes à maîtriser que les *Basic*, mais ô combien plus puissants.
Ils vous ouvrent les portes de la programmation professionnelle ou semi-professionnelle, en même temps qu'un monde de création limité par votre seule imagination.



Choix des options de compilation du GFA Assembleur.



Seul l'assembleur permet de réaliser des programmes en 3D suffisamment rapides pour offrir une réelle qualité de jeu.

L'Assembleur

Le microprocesseur 68000, cerveau des Atari, possède son propre langage: le langage machine. C'est un langage très spécial qui dispose de nombreuses instructions orientées vers la manipulation de la mémoire, le contrôle des circuits électroniques du ST, les opérations mathématiques élémentaires, etc. Les instructions de ce langage sont des codes numériques. Chaque code a une signification propre, comme lire ou écrire un octet en mémoire, copier une zone mémoire, comparer deux chiffres entre eux, faire une addition de deux octets, etc. Un programme en langage machine est une longue série de chiffres. Il existe tellement d'instructions dans ce langage qu'il est pratiquement impossible au commun des mortels d'écrire un programme en langage machine. Heureusement, il existe un moyen pour faciliter une telle programmation: l'*Assembleur*.




```
11:37  Insert I  Ligne : 1        Colonne: 1     ATARI.IS
; Appel par MACRO des fonctions du système d'exploitation
; (GEMDOS, BIOS, XBIOS)
; (C) November 1988 Peter Holzwarth


; Trap-Definitions


        .MACRO gemdos funktion,len      ; GEMDOS-Trap
        move.w   #\1,-(sp)              ; numéro de fonction
        trap     #1                     ; exécuter Trap
        .IF \?2  ;                      ; Transférer Parametre 2
        .IF \2<=8 ;                     ; oui
        addq.l   #\2,sp                 ; Ranger qq paramètres dans la pile
        .ELSE
        lea.l    \2(sp),sp              ; Ranger paramètres dans pile
        .ENDIF
        .ELSE    ;                      ; non
        addq.l   #2,sp
        .ENDIF
        .ENDM
```

Editeur du GFA Assembleur avec un exemple de source. Cet assembleur est l'un des plus puissants existant sur ST. Il n'est possible que d'ouvrir un seul fichier à la fois.

L'*Assembleur* est un langage du même type que le langage machine, mais où les codes numériques ont été remplacés par des symboles alphanumériques (lettres + chiffres). Par exemple, l'instruction 12 123 56 s'écrit `move.w 56,d0` (avant qu'un spécialiste du langage machine ne se mette à hurler, nous précisons que la séquence numérique est sortie de l'imagination de l'auteur de ces lignes). Cette dernière instruction est plus facile à retenir. Elle signifie que le microprocesseur doit écrire la valeur 56 dans la variable d0. Le symbole `w` indique que le nombre 56 est de type `Word`, c'est-à-dire qu'il occupe 16 bits.

Pour écrire un programme en langage machine, le programmeur commence par taper un listing source en *Assembleur*. Il utilise ensuite un convertisseur de code qui se nomme lui aussi, *Assembleur*. Celui-ci prend le listing puis génère un programme en langage machine. Cette transformation se nomme l'assemblage. Elle s'effectue généralement en deux opérations que l'on nomme passes d'assemblages. Il est possible que l'utilisateur ait fait une erreur de saisie en tapant son programme source. Dans ce cas, lors de l'assemblage, l'*Assembleur* affiche un message. Seules les erreurs évidentes, comme un nom d'instruction mal écrit, sont repérables par le programme. Les erreurs comme les inversions de variables ou les mauvaises valeurs numériques sont indécelables et causent des bogues.

Le microprocesseur 68000 ne possède que quelques variables appelées registres. Certains registres ont des fonctions particulières. Etant donné le petit nombre de variables, les données du programmes sont stockées dans la mémoire. Pour réaliser la moindre opération, il faut écrire une série d'instructions qui sauvent le contenu des registres, lisent une ou plusieurs données en mémoire, les chargent dans des registres spécialisés, effectuent un traitement spécifique et conservent le résultat en mémoire. C'est beaucoup plus compliqué qu'avec un langage évolué, où il suffit de taper, par exemple, a=b+c pour faire exactement la même chose.

Les avantages de l'Assembleur

Un programmeur qui connaît bien l'*Assembleur* peut écrire des programmes parfaitement optimisés qui s'exécutent extrêmement rapidement, tout en ne prenant que très peu de place en mémoire. Cette raison provient, en grande partie, du fait que l'*Assembleur* permet un contrôle totale des ressources du ST.

Les inconvénients de l'Assembleur

Il est très difficile d'apprendre à programmer en *Assembleur*. Un programmeur néophyte qui apprend le *Basic* peut commencer à écrire de petits programmes intéressants en deux jours. Le même programmeur qui étudie l'*Assembleur* ne peut commencer à écrire qu'au bout de deux ou trois mois (s'il n'a pas abandonné avant!). Une partie de la difficulté vient du fait qu'il faut bien connaître la structure interne du ST pour pouvoir écrire un programme intéressant.

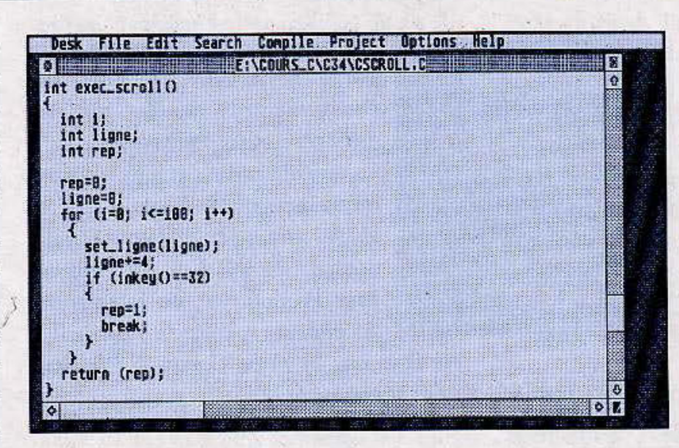
La programmation en *Assembleur* prend beaucoup de temps. Les spécialistes estiment qu'une équipe travaillant avec ce langage mettra quatre à cinq fois plus de temps pour développer le même logiciel qu'une équipe travaillant avec un langage de type *C*. Etant donné que le temps de développement moyen d'un logiciel professionnel est d'un an, on voit ce que cela peut donner!

De plus, il est très difficile de déboguer un programme en *Assembleur*, car la moindre erreur peut «planter» la machine. Cet problème augmente encore considérablement la durée de développement.

Etant donné que toutes les opérations, même les plus simples, comme une addition ou une soustraction, sont programmées par une suite d'instructions élémentaires, les programmes en *Assembleur* comptent un nombre de lignes incroyable. Il faut une centaine de lignes pour effectuer ce qu'un programme en *Basic* ferait avec une dizaine de lignes seulement!

Il existe une autre difficulté qui limite l'utilisation de l'*Assembleur* dans le domaine professionnel, c'est la «portabilité» des programmes. Un programme entièrement écrit en *Assembleur* est beaucoup plus difficile à transférer sur une autre machine qu'un programme écrit dans un langage évolué du type *C* ou *Pascal*.

Les seuls types de données que l'*Assembleur* connaît sont l'octet, l'entier court 16 bits et l'entier long 32 bits. Chaque fois que le programme doit manipuler d'autres types de données, comme des chaînes de caractères, il faut écrire des routines spécifiques. Là encore, c'est une grosse contrainte.



Editeur du Turbo C ST version 1.1. C'est l'environnement de développement le plus perfectionné et le plus performant qui soit sur Atari. Il est possible d'ouvrir plusieurs fenêtres simultanément.

Utilisation de l'Assembleur

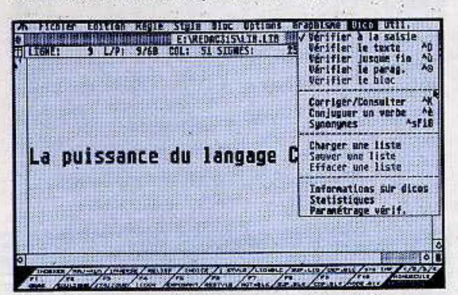
L'*Assembleur* est utilisé lorsque les besoins de vitesse et de place mémoire sont primordiales. La quasi-totalité des jeux d'arcade sont écrits en *Assembleur*. Bon nombre de jeux d'aventure et de programmes professionnels utilisent eux aussi des routines en *Assembleur* pour gérer les points critiques comme les routines graphiques. Les langages de programmation sont souvent écris en *Assembleur*.

Les Assembleurs ST

Les langages *Assembleur* le plus souvent utilisés sur ST sont *Devpack* de **Hisoft** et le *GFA Assembleur* de **Micro Application**. Ils ont été utilisés pour écrire presque tous les jeux d'arcade existant sur le marché.

Le langage C

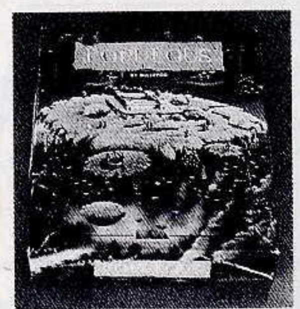
Le langage *C* est un langage moderne, développé dans les années soixante-dix par une équipe de recherche des **Laboratoires Bell** (aujourd'hui société ATT). Il a été conçu pour écrire un système d'exploitation (Unix) multitâche et «portable» sur de nombreuses machines. Les développeurs n'ont pas tardé à s'apercevoir des possibilités de ce langage et à l'utiliser d'abord comme langage système, puis comme langage multi-usage. Actuellement, la majorité des logiciels professionnels sont développés en *C*.



La puissance du langage C a été exploitée au maximum dans la programmation du célèbre traitement de texte Le rédacteur.



Un jeu écrit partiellement en C et en Assembleur. La partie écrite en C gère tout ce qui est de l'ordre de la simulation, alors que la partie Assembleur réalise les graphismes et les animations.



Les avantages du C

Le *C* a été créé pour travailler de manière modulaire, et ce afin d'utiliser les techniques de programmation structurée. Si le source est bien optimisé, la vitesse d'exécution des programmes *C* se rapproche souvent de celle de l'*Assembleur*.

Le *C* peut manipuler de nombreux types de données. Il possède même un mécanisme lui permettant de définir de nouvelles données appelées variables structurées. Ces dernières contiennent plusieurs types de données dans la même variable. Par exemple, on peut créer une variable contenant un nom, une adresse et un numéro de téléphone. C'est une technique qui facilite l'écriture de programmes lisibles et performants.

Pratiquement tous les ordinateurs possèdent un compilateur *C*, du ZX 81 au dernier modèle de Cray, en passant par toutes les machines 16 bits. On transfère ainsi facilement un programme sur une autre machine.

Il existe une importante bibliographie française sur le langage C. Les journaux spécialisés, y compris Atari Magazine, publient régulièrement des articles sur le sujet. De plus, le domaine public contient un grand nombre de programmes *C* intéressants, comme des compilateurs, des logiciels scientifiques, des programmes de gestion, des jeux, etc. Vous pourrez même trouver en téléchargement sur le 3615 ATARI le source *C* complet d'un compilateur C.

Les inconvénients du C

Pour exploiter pleinement les capacités du langage *C*, il faut comprendre un certain nombre de finesses et de concepts qui ne sont pas évidents au premier abord, d'où un apprentissage plus complexe.

Utilisation du langage C

Le *C* est le langage Atari par excellence. Voici quelques exemples de programmes écrits en C: les premières versions de GEM, le logiciel de dessin *Degas Elite*, la plupart des jeux d'aventure et de simulation, le traitement de texte *Rédacteur III*, de nombreux logiciels de gestion, plusieurs programmes de mise en page, etc. *Donjon Master* a été écrit partiellement en *C* et en *Assembleur*.

La majeure partie des logiciels commercialisés pour le Portfolio ont été conçus avec un compilateur *C* fonctionnant sur PC. L'un des meilleurs *Basic* Portfolio est un interpréteur du domaine public: le *PBasic*. Il a été écrit avec le *Turbo* C version PC (voir Atari Magazine n°32).

Les compilateurs C

Il existe de nombreux compilateurs C sur ST. Les plus utilisés sont le *Laser C*, le *Lattice C*, le *Mark & Williams C*, le *Megamax C* et le *Turbo C*. Le plus performant est le *Turbo C version 2.03* qui, pour des raisons assez complexes, n'est pas officiellement commercialisé en France. Cependant, les auteurs du compilateur ont créé une société qui vend le logiciel sous le nom de *Pure C*.

Ce dernier est très performant et possède un excellent environnement de développement, y compris un bon éditeur de ressource. De plus, à la différence de ses concurrents, il dispose d'une bibliothèque graphique très complète. De nombreux développeurs Français font appel à ce langage. On trouve la plupart des compilateurs *C* chez tous les distributeurs, excepté le *Pure C* (ou *Turbo C 2.03*) qui est uniquement importé en France par **Euromatique Technologie**.

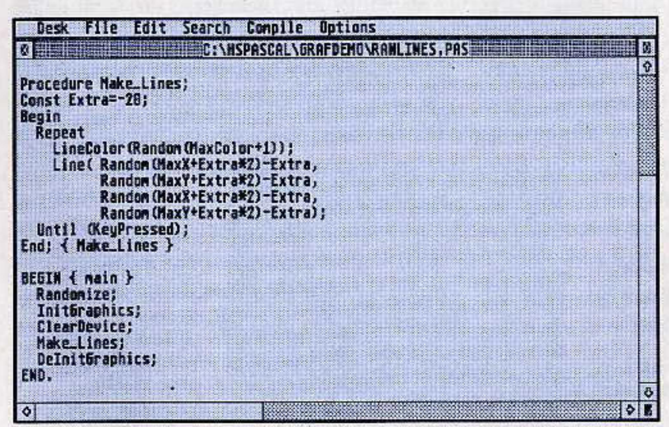
Vous trouverez sur le 3615 ATARI un compilateur C du domaine public: le *Sozobon C*. Il peut compiler n'importe quel programme C, bien que ses performances, au niveau vitesse et taille mémoire des programmes, soient inférieures à celles de ses concurrents commerciaux. Curieusement, ce compilateur est lui-même écrit en C. Ses sources sont également disponibles en téléchargement, et les articles publiés dans Atari Magazine s'appuient sur ce C peu coûteux.

Le Pascal

A l'origine, le *Pascal* est un langage conçu par un universitaire Suisse pour faciliter l'apprentissage de la programmation structurée. Il est devenu rapidement le langage privilégié de la communauté universitaire. De nombreux étudiants ont appris l'informatique avec le *Pascal*.

Les avantages du Pascal

C'est un langage simple qui s'étudie relativement vite, pratiquement aussi vite que le *Basic*. Un débutant pourrait s'initier à la programmation en commençant directement avec le *Pascal*. Malgré tout, c'est un langage puissant, intermédiaire entre le *GFA Basic* et le C, qui permet d'écrire des programmes performants. Du point de vue technique, c'est un langage structuré qui ressemble au C sur de nombreux points. Il permet d'écrire des programmes par assemblage de modules élémentaires, il gère de nombreux types de données et peut en définir de nouveaux. Un programme *Pascal* peut facilement être converti en C, alors que l'inverse est moins évident.

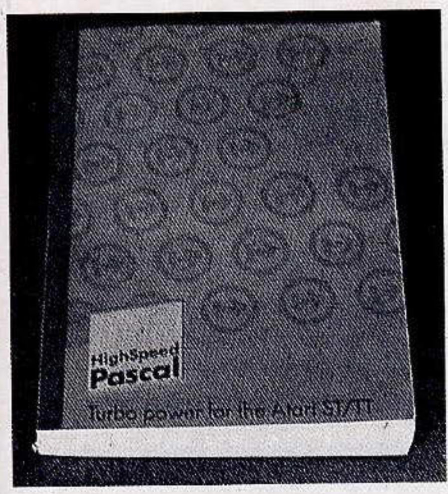


Editeur du HS Pascal. C'est un environnement de développement très agréable à manipuler. Ce produit est un clone presque parfait du célèbre Turbo Pascal Borland. Il est possible d'éditer plusieurs textes simultanément en ouvrant diverses fenêtres.

Les inconvénients du Pascal

Le seul inconvénient que l'on puisse trouver

Nom	Adresse	Téléphone
Euromatique Technologie	BP 60, 33032 Bordeaux Cedex	(16) 56 92 03 02
Dia Informatica	4, rue Joubert, 26000 Valence	(16) 75 44 21 25
Institut Pascal	26, Rue Lamartine, 75009 Paris	(1) 42 85 10 82



au *Pascal*, c'est qu'il n'est pas beaucoup utilisé sur ST. Il n'existe, sur ST, aucun ouvrage sur la programmation en *Pascal* et les journaux spécialisés ne publient pratiquement jamais d'articles sur le sujet. Cela vient de la concurrence effrénée du *GFA Basic* et du langage C. Le *GFA Basic* est très performant et convient très bien pour l'initiation. Les programmeurs qui veulent aller plus loin passent directement au C, alors que le *Pascal* est plus facile à apprendre. Cette constatation serait certainement différente s'il y avait eu un bon compilateur *Pascal* disponible peu après la sortie du ST, ce qui pas été le cas. Les premiers *Pascal* étaient relativement médiocres.

Utilisation du Pascal

Peu de logiciels ST ont été écrits en *Pascal*. Seuls quelques programmeurs de jeux d'aventure s'y sont essayés. Les programmeurs qui désirent aller plus loin que le *Basic*, sans affronter les complexités du C ou de l'*Assembleur*, devraient trouver leur bonheur dans le *Pascal*. Pour résumer, disons que le C est un langage professionnel, alors que le *Pascal* est un langage pour amateurs éclairés.

Les compilateurs Pascal

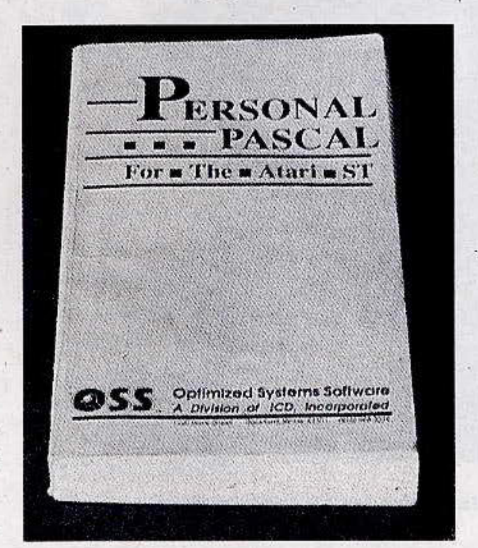
Les principaux compilateurs qui existent sur ST sont le *ST Pascal*, le *Pascal OSS* et le *HS Pascal*.

Le *HS Pascal* est le plus performant. Il possède un environnement de développement sous GEM facile d'utilisation, dont un éditeur multifenêtre très pratique. Sa vitesse de compilation est rapide et il utilise un système de bibliothèque de routines précompilées qui accélère encore la vitesse de compilation. Alors que les autres *Pascal* ne disposent pas de bibliothèque graphique, le *HS Pascal* version 2.0 en possède une très complète. Au point de vue efficacité et performance, il ressemble beaucoup au *Turbo Pascal PC* qui est l'un des logiciels les plus vendus au monde. On trouve les *Pascal* chez tous les distributeurs, sauf le *HS Pascal* qui est exclusivement importé par **Euromatique Technologie**. Attention: le *HS Pascal* est un produit danois importé en Allemagne, en Suisse et en Autriche par la société **Maxon**. Il existe donc aussi sous le nom de *M Pascal* (M comme **Maxon**). La documentation est en anglais et non en allemand comme on aurait pu le craindre. En partant de son premier langage, le concepteur du *Pascal* a développé un autre langage: le *Modula II*. Ce dernier ressemble beaucoup au *Pascal*, mais présente de nombreuses améliorations, notamment la possibilité d'écrire des programmes intégrant des routines fonctionnant en mode multitâche. A part le mode multitâche, toutes les améliorations du *Modula II* ont été repris dans les *Pascal* récents comme le *HS Pascal*. Il existe un compilateur *Modula II* du domaine public disponible en téléchargement sur le 3515 ATARI.

On peut trouver de nombreuses informations sur le *Pascal* à l'**Institut Pascal**. Cette boutique est spécialisée dans tout ce qui concerne ce langage (livres, revues, etc.), et abrite également un club d'utilisateurs.

Le Forth

Le *Forth* est un langage très particulier, mélange d'*Assembleur* et de langages évolués. C'est à la fois un langage interprété et un langage compilé. Il est beaucoup plus utilisé en Allemagne et dans les domaines de l'astronomie et de l'industrie. Les premiers jeux d'arcade Atari sur consoles étaient programmés en *Forth*. Il existe même des microprocesseurs dont le langage machine est directement en *Forth*. Ils sont essentiellement utilisés dans l'industrie et dans les applications militaires.



Le *Forth* fonctionne autour du concept de la notation polonaise inverse (comme les calculatrices scientifiques de type HP), et d'un noyau d'instructions écrit en *Assembleur*. L'utilisation de ce langage est très différente des autres. Avec un langage classique, le programmeur écrit un listing source qui est soit compilé, soit exécuté par un interpréteur. Avec le *Forth*, le programmeur travaille de manière interactive et définit de nouvelles instructions à partir des instructions du noyau de base.

Ces dernières sont immédiatement compilées dans un code proche du langage machine. Le noyau de base contient de puissantes instructions ressemblant beaucoup aux instructions de l'*Assembleur*.

C'est un langage puissant, mais complexe. Pour l'utiliser pleinement, il faut appréhender des principes différents de ceux qui ont cours dans l'informatique classique. Les programmeurs *Forth* forment une sorte de «secte» par rapport au reste du monde informatique.

Les Forth sur ST

Il existe deux *Forth* du domaine public sur le 3615 Atari. Le meilleur des deux est certainement le *Forth 83* qui suit une norme édictée par le **Comité de normalisation du Forth**.

Le Lisp et le Prolog

Le *Lisp* et le *Prolog* sont des langages destinés à développer des programmes d'intelligence artificielle. Ils fonctionnent selon des principes très différents des autres langages.

Les programmes écrits en *Lisp* sont des listes d'objets possédant différentes propriétés. Les programmes *Prolog* ressemblent plus à des énoncés de problèmes mathématiques qu'à des programmes traditionnels. Les seules personnes susceptibles d'être intéressées sont les étudiants en informatique, qui les utilisent pour faire chez eux leurs travaux pratiques, et les spécialistes vraiment pointus.

La société **DIA Informatica** commercialise un *Lisp* et un *Prolog*. Vous trouverez sur le 3615 ATARI un *Lisp* du domaine public: le *XLisp*. Ce programme est écrit entièrement en langage C. Le fichier téléchargé contient le programme exécutable, mais aussi les sources C.

Patrick Leclercq ■

PROFESSION PROGRAMMEUR
La passion avant tout

La programmation peut aussi devenir un métier. Nous avons rencontré deux développeurs qui exercent dans des domaines différents: le jeu et le professionnel. Ils nous livrent leurs secrets.

Hervé Lange et Cyril Saxstad ont chacun choisi de devenir programmeur professionnel. Bien que travaillant dans deux domaines différents, et même s'ils n'ont pas choisi les même langages de programmation, ils ont à peu près les même méthodes de travail. Peut-être sont-elles valables pour l'amateur que vous êtes.

La passion du jeu

Hervé Lange a participé à la création du jeu d'aventure *BAT* édité par **Ubi Soft.**

Atari Magazine. L'informatique a-t-elle tenu une place prépondérante dans votre formation?

Hervé Lange. Je suis titulaire d'un Bac F2 et d'un DUT spécialisé dans les microprocesseurs. Ma formation de base est essentiellement axée sur l'électronique. J'ai découvert la programmation par moi-même, comme beaucoup d'électroniciens.

A.M. Les connaissances acquises à l'école vous servent-elles dans le cadre de vos développements?

H.L. Elles me sont très utiles. Mes études m'ont permis d'acquérir une méthode de travail qui m'aide énormément. Pour écrire un bon logiciel, il faut surtout réfléchir. Mes meilleurs outils sont encore un papier et un crayon.

A.M. Vous avez collaboré à la création de plusieurs logiciels, quels sont-ils?

H.L. Sur Amstrad CPC, j'ai participé à la création des logiciels Fer & Flammes et L'Anneau de Zengara. Sur ST, j'ai collaboré avec l'équipe de Computer's Dream pour créer les jeux d'aventure BAT et BAT II, ainsi que l'utilitaire Music Master ST.



L'équipe de programmation de Computer's Dream en compagnie de l'attachée de presse d'Ubi Soft.

A.M. Vous avez choisi une organisation en équipe, parlez-nous de Computer's Dream.

H.L. Computer's Dream est un groupe de 8 développeurs indépendants qui réalisent des logiciels de jeux. Nous avons des programmeurs, des graphistes et des sonorisateurs. L'ère du programmeur génial qui écrit un soft fabuleux dans sa chambre est passée depuis longtemps. Les logiciels actuels sont tellement complexes que seule une équipe de plusieurs personnes peut en venir à bout dans un temps raisonnable.

A.M. Quel rôle jouez-vous au sein de Computer's Dream?

H.L. Mes tâches sont multiples. Je suis designer/scénariste, ainsi que programmeur. Par exemple, j'ai écrit le scénario de BAT II, mais je suis aussi chef de projet pour les versions Amiga et PC.

A.M. Une des questions que l'on se pose souvent est de savoir combien gagne un programmeur comme vous?

H.L. C'est extrêmement variable. En 1990, je gagnais à peu près 9 000 F net par mois. En 1991, environ 15 000 F. A l'heure actuelle, je ne pense pas toucher plus de 9 000 F net par mois.

A.M. Cela n'est-il pas dérisoire par rapport au succès de BAT?

H.L. Il ne faut pas oublier que l'équipe de Computer's Dream compte huit personnes. Nous sommes parmi les développeurs français qui gagnent le plus d'argent. Beaucoup de programmeurs de jeux touchent le smic, voire moins. La profession de créateur de jeux n'est pas très rentable pour l'instant.

A.M. Quelles sont vos motivations premières?

H.L. Essentiellement, la joie de travailler sur

un sujet qui me plaît et le plaisir de la création. Les considérations financières ne viennent qu'après, bien que je sois convaincu que le domaine du jeu vidéo va connaître une grande expansion dans les années à venir. Comme les pionniers du cinéma au début du siècle, nous créons actuellement des produits et des concepts qui seront dans quelque temps accessibles au grand public.

A.M. Utilisez-vous un ou plusieurs langages pour écrire vos programmes?
H.L. BAT comprend environ 50% de GFA Basic et 50% de langage machine. BAT II est entièrement écrit en Assembleur. Il compte environ 300.000 lignes d'Assembleur. Nous avons utilisé le GFA Assembleur qui est un excellent produit mis à part quelques défauts mineurs. Malgré la tendance actuelle, je ne suis pas partisan d'utiliser le langage C, même en y intégrant des routines en assembleur. Je pense qu'il est préférable d'écrire un jeu totalement en assembleur.

A.M. Quel est le processus de développement d'un programme? A combien évaluez-vous la phase de développement proprement dite?
H.L. Le meilleur exemple est celui de BAT II. Nous avons commencé par chercher une idée, puis nous avons réfléchi dessus pendant deux mois. Notre principe est de prendre un concept intéressant et d'en faire une adaptation informatique comme on pourrait en faire une adaptation pour le cinéma. La phase de développement proprement dite nous a pris dix-huit mois. Grâce à la bonne conception initiale, la phase de déboguage n'a duré qu'un mois. C'est exceptionnel pour un logiciel de cette taille. La phase de finition a pris environ deux mois. Dans l'ensemble, nous avons dépassé nos prévisions de six mois. C'est un chiffre raisonnable par rapport aux retards moyens des éditeurs français et américains. Il existe d'ailleurs un «proverbe informatique» qui dit que le temps de développement d'un programme est toujours supérieur de 20% aux prévisions.

A.M. Qu'appelez-vous phase de finition?
H.L. Cette phase est consacrée à l'étude de la jouabilité, à l'optimisation de certains algorithmes et à la rédaction d'un dossier technique destiné à l'éditeur. Les éventuelles adaptations du logiciel seront réalisées à partir de ce dossier. La phase de finition est une phase critique, car c'est souvent à ce niveau que les développeurs craquent.

A.M. Le travail a-t-il été entièrement effectué par le groupe Computer's Dream?
H.L. Non! Nous avons employé des intervenants extérieurs pour accomplir des tâches ponctuelles. Par exemple, certains dessins ont été faits par des graphistes extérieurs rémunérés à cet effet.

A.M. Est-ce que ces intervenants extérieurs vous posent des problèmes particuliers?
H.L. La plupart des graphistes ont du mal à se soumettre aux contraintes découlant du développement d'un jeu, notamment les impératifs de qualité. Ils ont tendance à ne pas respecter le style graphique du jeu et à utiliser leur propre style. Cela nuit à la cohérence d'ensemble du logiciel. De plus, ils se plient mal aux limitations techniques et dessinent des images trop grandes, ou utilisent une palette de couleur différente de celle du jeu. Ceci dit, avec un peu d'expérience, ils font de l'excellent travail.

A.M. Le coût de ce genre d'opération doit être relativement élevé...
H.L. Développer actuellement un jeu de bonne qualité coûte énormément d'argent. Il faut rémunérer les intervenants extérieurs, payer l'équipe de développement qui ne peut pas vivre pendant deux ans sans argent, acheter l'équipement (machines, langages, documentations, etc.) Une somme de 300 000 F représente un petit budget de développement. Le budget de BAT II est d'environ 800 000 F, pour un jeu développé simultanément sur 3 machines.

A.M. Est-il facile de créer des logiciels destinés à être vendus dans plusieurs pays?
H.L. C'est très difficile. Chaque pays a ses habitudes particulières et les importateurs locaux insistent pour que les jeux respectent certaines règles de moralité. Certaines jeunes femmes apparaissant dans BAT étaient peu vêtues. Elles ont dû être complètement rabillées pour la version allemande. Toujours dans la même version, le mot «ennemi» a été changé par l'expression «non-ami». Pour les Américains, la «maison de passe» a été remplacée par «salle de danse». La philosophie des jeux n'est pas la même selon les pays. Les Américains aiment les choses simples, mais qui durent longtemps. Les Japonais adorent les choses compliquées et la violence ne leur posent pas de problème.

A.M. Lors de la création d'un programme, quel est le critère le plus important?
H.L. C'est sans conteste la jouabilité. Les jeux doivent être simples à utiliser. La documentation ne doit pas être trop épaisse pour ne pas décourager le joueur. L'environnement sonore est aussi très important. Les développeurs qui viennent du monde des 8 bits ont tendance à négliger l'aspect sonore au profit du graphisme. C'est une erreur que j'ai moi-même commise et que j'ai tenté de rectifier avec BAT II.

A.M. Vous avez sans doute quelques conseils à donner à un développeur qui veut se lancer dans la création d'un jeu.
H.L. Il faut d'abord savoir que le milieu de l'édition du jeu est impitoyable. Ensuite, il ne faut pas commencer un jeu sur une technique spéciale comme une routine de scrolling. Il faut bien différencier la technique et le scénario. Un programme techniquement superbe peut être sans intérêt, alors qu'un programme techniquement banal, avec un bon scénario, peut être du plus haut intérêt. Le public ne s'intéresse pas à la performance technique, mais à son amusement. Il faut savoir faire des concessions pour répondre aux demandes des éditeurs et des personnes qui vont collaborer au développement. Etant donné la masse de travail exigée par la réalisation d'un soft de jeu, l'appât du gain ne doit pas être la seule motivation, sous peine de risquer d'abandonner le projet. Il faut faire vraiment beaucoup d'efforts et être prêt à sacrifier ses fins de semaines et parfois ses vacances. Il faut aussi regarder ce qui se fait ailleurs pour ne pas le refaire.

A.M. D'après vous, quel est l'avenir de l'informatique de jeu?
H.L. C'est très difficile à prévoir. On peut cependant remarquer que les jeux deviennent de plus en plus gros et, bientôt, ne fonctionneront plus que sur disque dur ou CD-ROM. Au sein de Computer's Dream, nous orientons essentiellement nos recherches selon deux axes: l'intelligence artificielle et la 3D. Ce sont des éléments qui, dans l'avenir, compteront certainement de plus en plus.

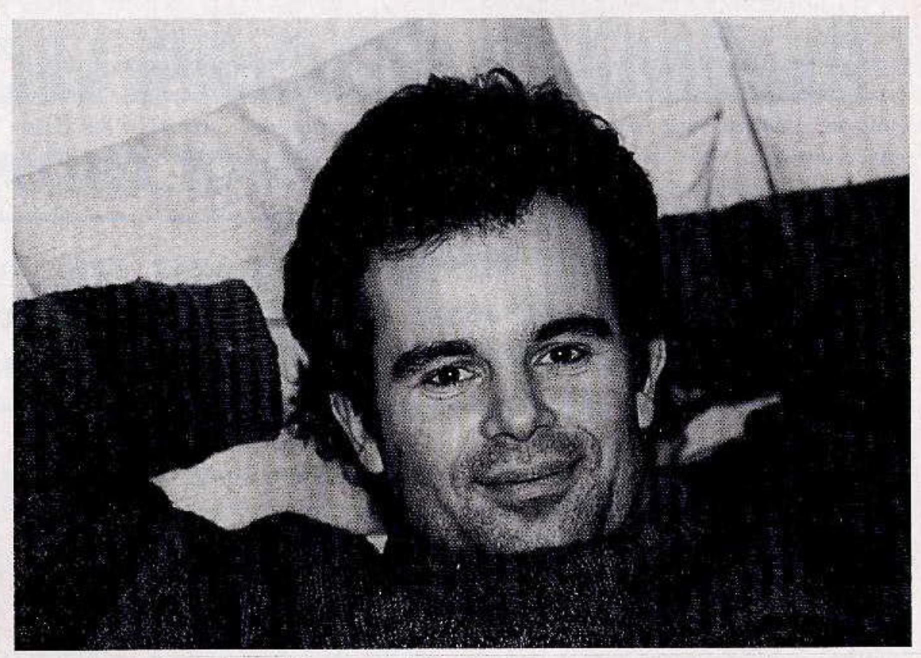
A.M. Quelle place les Français occupent-ils sur le marché mondial de l'édition?
H.L. J'ai l'impression que les éditeurs anglais et américains prennent de plus en plus d'importance par rapport aux éditeurs français. C'est dommage, car nous étions bien partis. Il faut plus d'équipes de développeurs Français, car la culture d'origine des déve-

loppeurs a une influence sur la conception du jeu. Il y a de la place pour de nouveaux développeurs à condition que ces derniers se comportent comme des professionnels et non comme des amateurs.

L'utilisateur avant tout

Cyril Saxstad, jeune homme d'origine franco-norvégienne, a réalisé de nombreux logiciels professionnels sur ST. Il a bien voulu nous expliquer certaines de ses techniques de développement.

Atari Magazine. Votre formation vous a-t-elle directement amenée à côtoyer le monde de l'informatique?
Cyril Saxstad. J'ai fait des études supérieures et je possède une maîtrise de biochimie. En revanche, tout ce que je sais en informatique, je l'ai appris par moi-même.


Cyril Saxstad, ardent défenseur du Turbo C.

A.M. Vous êtes à l'origine de plusieurs logiciels?
C.S. J'ai écrit GesStck 90, un logiciel de gestion de stocks et de facturation; Compta 91, un logiciel de comptabilité; GesBar qui permet d'imprimer des codes barres et La Cuisine.

A.M. Quels sont les revenus annuels d'un programmeur de logiciel professionnel?
C.S. En 1991, les logiciels ST m'ont rapporté 80 000 F. Cela représente environ un tiers de mes revenus. Le reste provient de développements réalisés sur d'autres machines.

A.M. Avez-vous des préférences pour certains langages?
C.S. Mes premiers logiciels ont été écrits en GFA Basic avec des routines en Assembleur. L'Assembleur servant uniquement à gérer les points critiques comme les effets spéciaux graphiques. Je travaille actuellement en Turbo C.

A.M. Pourquoi programmer en langage C?
C.S. C'est le seul langage qui permette de développer correctement des applications importantes utilisant les ressources du ST et gérant convenablement la mémoire. La gestion mémoire est un point important à ne pas négliger. La possibilité de créer des variables structurées est très importante. Elles clarifient la pensée et permettent d'écrire des programmes plus lisibles.

A.M. Est-ce que vous utilisez des routines en Assembleur dans vos programmes en langage C?
C.S. J'utilise toujours l'Assembleur pour traiter certains problèmes particuliers comme la décompression des données graphiques, les effets spéciaux graphiques et les manipulations d'images en général.

A.M. A cet effet, quel Assembleur utilisez-vous?
C.S. J'utilise DevPack ST.

A.M. Avez-vous rencontré des problèmes de compatibilité entre les différents types de ST?
C.S. Je cherche toujours à écrire des programmes «portables» (utilisables sur différents ordinateurs) en me servant des fonctions du système d'exploitation Atari. Cela complique le travail, mais permet d'obtenir des programmes parfaitement «portables». Je n'ai eu aucun problème de compatibilité avec les différentes machines Atari, y compris avec les TT. J'ai même eu le plaisir de voir l'un de mes programmes fonctionner sur un écran géant en exploitant la totalité de cet écran. Il est payant de respecter les conventions de programmation dictées par Atari et de ne pas céder à la tentation de «bidouiller» pour optimiser un programme.

A.M. Quelles sont les différentes étapes sur lesquelles vous vous basez pour créer un logiciel?
C.S. Je commence par étudier les fonctionnalités du programme, ainsi que la structure de l'interface utilisateur. Cela prend entre un et deux mois. Le travail suivant est la définition de la structure des fichiers de données. C'est une étape importante pour obtenir un bon logiciel.
Je programme ensuite toutes les routines dont je vais avoir besoin. Cela simplifie l'écriture du programme et me permet de constituer des bibliothèques de fonctions réutilisables dans d'autres programmes. Il arrive souvent que pendant la phase de programmation je sois amené à modifier la conception initiale. Ce travail de programmation prend entre trois et quatre mois. Je m'occupe ensuite du déboguage. C'est une phase qui s'exécute assez rapidement grâce à un code source propre, une programmation modulaire et une bonne organisation des données. Le logiciel est ensuite confié à des clients particuliers qui me font part de leurs remarques et des bogues qu'ils détectent. Cette phase de finition prend un mois. La réalisation complète d'un programme me prend entre six et huit mois.

A.M. Prenez-vous beaucoup de temps pour écrire la documentation de vos programmes?
C.S. La documentation est un élément important d'un programme, puisque c'est le document de base de l'utilisateur. Elle est souvent négligée par les programmeurs qui s'en occupent au dernier moment. J'essaie de garder suffisamment d'énergie pour faire convenablement ce travail. Il me faut environ deux semaines pour écrire une bonne documentation.

A.M. Quel est, selon vous, l'élément le plus important dans un programme?

C.S. C'est sans conteste l'interface utilisateur. Les auteurs de logiciels ne doivent pas oublier qu'ils ne travaillent pas pour leur propre plaisir, mais pour des utilisateurs.

Un logiciel puissant avec une interface pauvre est un programme moyen. Un autre logiciel avec des fonctionnalités moindres, mais une excellente interface utilisateur est bien meilleur.

Il est préférable de se consacrer à la conception d'une bonne interface plutôt que d'ajouter quelques fonctionnalités supplémentaires qui ne serviront pas à tout le monde. Lorsque je conçois un programme, je ne cesse de penser à l'utilisateur final.

A.M. Les utilisateurs sont-ils sensibles à la qualité de ces interfaces?

C.S. La perception de la puissance d'un programme par un utilisateur dépend de sa simplicité d'utilisation.

Un programme peu puissant, mais avec une très bonne interface sera considéré comme meilleur qu'un programme plus puissant avec une mauvaise interface, car l'utilisateur aura moins de problèmes pour exploiter entièrement le programme.

A.M. Comment réalisez-vous vos interfaces utilisateur?

C.S. Toutes mes interfaces utilisateur ont été réalisées avec l'utilitaire K-ressource.

J'utilise les ressources GEM car cela permet de développer très rapidement une interface, tout en ayant une vision globale du résultat final. La conception est beaucoup plus simple et il est possible de faire des modifications à n'importe quel moment.

On peut alors réaliser une version étrangère d'un programme en ne modifiant que son fichier ressource. De plus, comme les ressources sont gérés par GEM, les programmes qui les utilisent fonctionnent sur n'importe quelle version de GEM, diminuant ainsi les problèmes de «portabilité».

A.M. Apparemment, vous avez une certaine réticence vis à vis des interfaces utilisateur ST?

C.S. Les programmes ST souffrent d'une non-homogénéité de leurs interfaces utilisateur. Tous les programmeurs conçoivent leurs interfaces sans tenir compte du travail des autres.

Cela donne parfois de beaux logiciels, mais à chaque nouveau programme, l'utilisateur doit réapprendre l'environnement dans lequel il va travailler.

Il est dommage qu'Atari n'ait pas publié des spécifications précises sur la présentation des interfaces utilisateur, comme d'autres constructeurs l'on fait.

A.M. A part l'interface utilisateur, à quels autres aspects du programme attribuez-vous de l'importance?

C.S. Les développeurs s'occupent du module d'impression au dernier moment et le traitent trop rapidement pour obtenir un bon résultat. C'est une grosse erreur car l'impression est une chose importante. Les programmes doivent produire des documents propres et parfaitement présentables.

En gérant les imprimantes avec GDOS, on obtient des résultats intéressants. Hélas, il y a un manque complet de documentation GDOS de la part d'Atari.

Il faut prendre au moins un mois pour en comprendre les subtilités, mais cela vaut la peine et permet d'obtenir des documents de bonne qualité. La sortie récente de FSMGDOS est intéressante, car il bénéficie d'une bonne documentation, tout en étant très performant et totalement compatible avec GDOS.

L'installation des logiciels est aussi une chose importante. J'équipe tous mes programmes avec un logiciel d'installation grâce auquel l'utilisateur définit son type de machine, la présence ou l'absence d'un disque dur et le type d'imprimante. Si un disque dur est présent, le logiciel s'installe automatiquement dessus. La tâche de l'utilisateur est alors simplifiée.

A.M. Quelles sont les documentations auxquelles vous vous référez le plus souvent?

C.S. La seule documentation que j'utilise régulièrement est «Le livre du développeur» de Micro Application. Pour les points particuliers comme les spécificités du TT ou FSMGDOS, j'utilise des sources internes à Atari.

A.M. Quels conseils donneriez-vous à un programmeur débutant?

C.S. Je lui conseille d'apprendre la programmation avec le GFA Basic 3.0. Malgré sa puissance, c'est un langage structuré, simple à utiliser et à mettre en œuvre. Il ne comporte pas de numéro de ligne et permet d'écrire des programmes relativement propres. De plus, c'est un langage très répandu qui comporte une importante documentation. Il existe de nombreux livres et un grand nombre d'articles sur la programmation en GFA Basic.

A.M. ...et à un programmeur déjà expérimenté voulant aller plus loin?

C.S. Je lui conseille de se mettre au langage C et d'apprendre à utiliser les fichiers de ressources GEM.

Propos recueillis par
Patrick Leclercq ■

LE PORTFOLIO ET SON ECRAN
De belles images sur votre portatif

L'écran du Portfolio est le plus souvent utilisé en mode texte, alors qu'il exploite parfaitement le mode graphique. Souvenez-vous qu'un beau graphique vaut souvent mieux qu'un long discours.

Le Portfolio est doté d'un écran LCD de 8 lignes x 40 colonnes en mode texte, et de 240x64 points en mode graphique. Ici, nous ne parlerons que du mode «standard», car les autres sont peu utilisés. En mode texte, le système s'occupe de tout. En revanche, pour ce qui est du graphisme, quelques points doivent être éclaircis.

Le logo Portfolio officiel

Technique

La taille de l'écran est donc de 240x64 points, soit 1920 octets. Le début de l'écran en mémoire est situé à l'adresse 0, segment B000 (en hexadécimal bien sur...).

En théorie, un simple Poke à cette adresse doit afficher quelque chose. Mais cela ne marche pas. Pourquoi? Tout simplement pour économiser les piles. En effet, pour éviter de devoir réafficher tout l'écran à chaque changement de point, les concepteurs ont décidé qu'il fallait utiliser une fonction spéciale du système pour transférer la mémoire écran dans l'écran lui-même. Cette fonction s'appelle `Screen Refresh` et fait partie de l'interruption 61H, service 12H, et il n'existe aucun paramètre en entrée (donc `INT 61H`, et `AH=12H`). Voilà pour les techniciens purs et durs. Pour les autres, ne vous en faites pas, les langages spécifiques au Portfolio (*PBasic*, *Lna*) tiennent compte de ces caractéristiques et vous proposent des instructions graphiques fonctionnant très correctement.

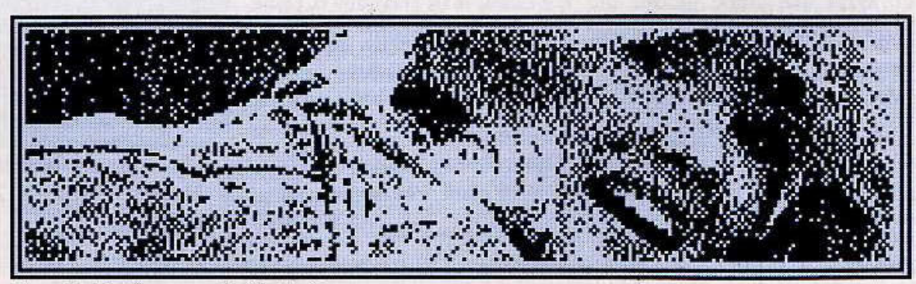
Le logo Atari

Des images bitmap

A part de beaux graphes, il n'y a pas encore (à notre connaissance) de programmes de dessin sur le Portfolio. En revanche, il est possible d'obtenir des images digitalisées sur le «petit écran» du Portfolio, tout simplement en transférant une partie d'un écran provenant de n'importe quel ordinateur de table (ST, PC, Mac, etc.). Un petit programme permet de découper une partie de l'écran au format Portfolio et de le sauver sur le disque. Ensuite, il ne reste plus qu'à le transférer avec le logiciel adéquat. Les formats d'écrans pour le Portfolio sont les suivants: `.PGF` (pour Portfolio Graphic File) et `.PGC` (pour Portfolio Graphic Compressed, fichier compressé par une méthode Pac-Byte). Pour les afficher sur votre petite machine, vous pouvez utiliser le programme *PGSHOW*, l'instruction `PGSHOW` du *PBasic*, les routines que nous vous fournissons en *LNA*, ainsi que n'importe quel langage permettant de «poker» et de lancer une interruption. Pour ce qui est du format PGF, les octets se suivent les uns derrière les autres. Pour le PGC, la méthode est disponible en téléchargement sur le **3615 ATARI**, ainsi qu'un programme de compactage pour le Portfolio.

Gérer l'écran en LNA

La souplesse du *Lna* permet (comme nous l'avons vu lors du précédent numéro) d'utili-



Une belle image digitalisée

ser efficacement l'assembleur pour ajouter de nouvelles instructions au langage. Le domaine du graphisme en tire énormément profit. Afin que vous puissiez afficher vos superbes images dans ce langage, nous vous proposons un module (IMAGE.MOD, à inclure dans vos programme) qui découpe et restitue dans un tableau des morceaux d'écran en mode graphique. Son fonctionnement est expliqué dans les lignes de commentaires.

La routine assembleur G_VIDEO.BIN est sous la forme d'un fichier hexadécimal (G_VIDEO.HEX), et nécessite, pour être codée, le programme HEXA_BIN du mois dernier. Pour vous familiariser avec ce module, nous vous proposons un petit programme d'exemple (IMAGE.LNA), qui affiche une image PGC du disque (en l'ayant décompacté avant!).

Des images du ST

L'utilisateur d'un ST dispose généralement d'une imposante bibliothèque d'images en tout genre, qu'il rêve de partager avec son portfolio. Que ce soit pour la représentation d'un programme ou tout simplement pour «épater la galerie».

Le programme *SnapPortfolio* vous permet de découper un morceau d'une image bitmap ayant l'une des extensions suivantes: DOO, PIC, PI3, SD2. Après le chargement d'une image, déplacez le rectangle (symbolisant l'écran du Portfolio) sur la partie de l'image qui vous intéresse, cliquez sur le bouton gauche pour sauver, ou sur le droit pour quitter. Lors de la sauvegarde, vous pouvez choisir entre le format normal (PGF) et le format compacté (PGC). Ce programme fonctionne en *GFA Basic*, à partir de la version 3, et uniquement en monochrome.

N'hésitez pas à le télécharger ou à commander la disquette sur lequel il se trouve (voir encadré en fin d'article).

Pour finir...

Ce programme n'est qu'un exemple. Libre à vous de l'adapter et de le transformer. Il serait également possible d'en faire un accessoire, très simplement...

Pour ceux qui n'utilisent pas un ST, d'autres programmes existent sur les différentes machines du commerce.

Quand à l'utilisation d'images sur le Portfolio, la place est laissée à votre imagination.

On pourrait, par exemple, voir apparaître des catalogues de produits ou de références agrémentés de schémas ou de photos digitalisées !

Bertrand Espern ■

Le Listing

L'abondance des sujets traités dans ce numéro ne nous permet pas d'imprimer les programmes correspondants. Pour vous les procurer, téléchargez les sur le 3615 ATARI ou procurez vous la disquette correspondante (voir bon de commande encarté à la fin du numéro).



TRANS-DRIVE
Le transfert à grande vitesse

Quatorze fois plus rapide que ses prédécesseurs, Trans-Drive deviendra vite indispensabl pour effectuer des transferts de fichiers entre un Portfolio et un PC de bureau.

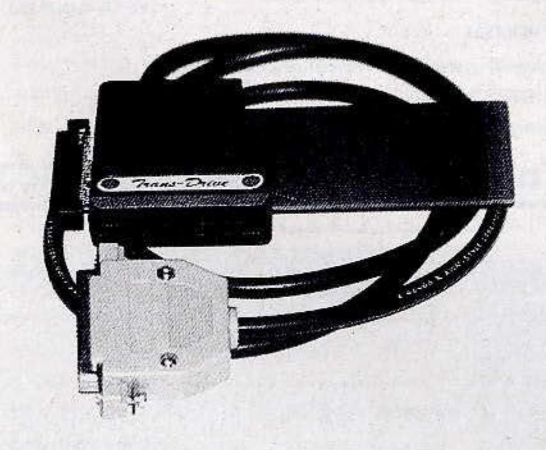
Le Portfolio est un ordinateur de poche idéal pour tous ceux qui sont amenés à effectuer de fréquents déplacements.

Il peut être utilisé pour saisir des données, taper des textes ou se servir de programmes spécifiques. De nombreux commerciaux l'utilisent afin de réaliser, directement sur le terrain, des calculs de devis. Les fichiers de données peuvent ensuite être transférés sur un PC ou un ST, pour traitements ultérieurs et archivage. La ROM du Portfolio contient un programme de communication utilisable avec le programme *FolioLink* sur ST ou le programme *FT* sur PC. Ce dernier (*FT*) est fourni avec l'interface parallèle Atari, et fonctionne sans elle. Ces deux systèmes étant assez lents, on ne peut guère s'en servir pour un travail sérieux.

Pour donner un ordre de référence, la vitesse de transfert est d'environ 380 octets à la seconde. Cela correspond approximativement à 22 Ko à la minute.

Trans-Drive est une extension qui accélère considérablement le transfert de vos fichiers puisqu'il fonctionne à la vitesse de 320 Ko à la minute. Ce produit est fabriqué par la société Allemande **Becker & Partner**.

Branchement sur le port beecard. Ce mode de connexion interdit l'utilisation d'une carte RAM mais permet d'accéder à la ROM du Trans-Drive.



Trans-Drive et son câble permettent la connexion avec le port parallèle du PC.

Le produit

L'exemplaire de *Trans-Drive* que nous avons testé nous a été fourni sous la forme d'une carte électronique, d'une disquette 5 pouces 1/4, 360 Ko, d'une disquette 3 pouces 1/2 720 Ko, et d'une documentation en allemand. Sur notre demande, la société **Becker & Partner** nous a faxé une documentation en français. La carte peut se connecter de deux manières différentes: sur le port beecard du Portfolio ou sur le port d'extension droit. Elle comprend un câble qui se branche sur le port imprimante du PC.

Installation

Trans-Drive possède sa propre ROM. Cette dernière contient des fichiers indispensables à la réalisation d'un transfert. Pour installer ces fichiers dans le Portfolio, il faut connecter la carte sur le port beecard et lire le disque `A:` selon les indications de la documentation.

Vous pouvez alors changer la carte de place et la connecter sur le port d'extension droit. Sinon, vous vous trouverez dans l'impossibilité d'utiliser une beecard, ce qui est bien dommage.

Nous avons eu un petit problème pour installer Trans-Drive car la documentation fournie par **Becker & Partner** correspondait à une ancienne version. Si vous avez le même problème, c'est-à-dire si vous ne trouvez pas les fichiers `CONFIG.SYS` et `TDP.SYS` sur la disquette `A:`, sachez que ces deux fichiers se trouvent dans le répertoire `C:\TDPA`, ou dans le répertoire `C:\TDPB`.

Utilisation de Trans-Drive

Les disquettes fournies avec Trans-Drive contiennent le programme `TDP.EXE` Lorsque ce dernier est lancé sur PC, le lecteur du PC devient le lecteur `D:` du Portfolio. Vous pouvez alors utiliser sur ce disque toutes les commandes DOS du Portfolio (`DIR`, `COPY`, `DEL`, `REN`, `CD`, `RM`, etc.), et même exécuter di-

rectement des programmes à partir du disque. Pour des raisons évidentes de sécurité, la seule commande DOS inutilisable est la commande FORMAT. Vous pouvez préciser quel lecteur PC qui deviendra le lecteur du Portfolio. Le transfert d'un fichier disque PC dans le RAM Disque C: du Portfolio s'effectue de la manière suivante:

```
COPY D:PRG.EXE C:
```

La communication dans le sens Portfolio vers PC se fait tout aussi naturellement avec l'instruction COPY.

```
COPY C:*.TXT D:
```

L'utilisation est aussi simple que si le lecteur de disquette se trouvait directement connecté au Portfolio. C'est beaucoup plus pratique que les autres systèmes de transfert qui nécessitent l'emploi du logiciel de transfert présent dans la ROM du Portfolio.

Les performances

Dès les premiers essais, nous avons été agréablement surpris par la rapidité de transfert. Un fichier de 16008 octets à été transféré en trois secondes. Afin d'avoir une valeur de référence, nous avons effectué cette opération avec le même fichier grâce au système fourni avec l'interface parallèle Atari. Celui-ci a mis quarante-deux secondes pour effectuer le travail. Trans-Drive est donc 14 fois plus rapide que le kit de transfert Atari. Sa vitesse est d'environ 320 Ko à la minute, contre 22 Ko pour le système Atari.

Remarque: le lecteur de disquette met un certain temps pour atteindre sa vitesse de travail. Ce temps de mise en route est d'environ trois secondes. Pour que cette période de temps ne perturbe pas le résultat des mesures, les fichiers n'ont pas été transférés à partir d'un lecteur de disquette PC, mais à partir d'un RAM Disque PC. Les tests ont été effectués avec un Portfolio 128 Ko équipé d'une beecard de 128 Ko, et avec un PC AT équipé d'un lecteur 5 pouces 1/4 1,2 mégas et d'un lecteur 3 pouces 1/2 1,44 mégas.

Comment fonctionne Trans-Drive?

Les premiers systèmes de communication entre ordinateurs utilisaient une liaison série RS 323. Ce mode de liaison consiste à transférer les bits les uns après les autres. La vitesse de transfert se mesure en bauds, c'est-à-dire en nombre de bits par seconde. Rappelons qu'un octet contient 8 bits. Beaucoup de liaisons séries se font à 1 200 bauds. Le défaut de cette technique, c'est que les bits sont transmis un par un. Afin d'accélérer la vitesse, il est possible d'utiliser l'interface parallèle de l'imprimante. Ce type de liaison envoie directement des octets et non plus des bits. Le kit de transfert d'Atari utilise l'interface parallèle, mais sa vitesse ne dépasse pas les 22 Ko par minute. D'après les techniciens de Becker & Partner, Trans-Drive atteint la vitesse de 38 000 bauds, ce qui correspond aux résultats de nos tests. Ceci dit, il est peut-être possible de faire encore mieux, puisque certains systèmes de transfert entre PC du type PC Link atteignent la vitesse fabuleuses de 115 000 bauds.



Trans-Drive se branche aussi sur le port d'extension du Portfolio. Ce type de connexion libère le port beecard et facilite l'utilisation d'une carte RAM pour stocker les fichiers transférés à partir du PC.

Sauvegarde de la mémoire

Puisque le lecteur PC est accessible par le DOS du Portfolio, vous pourrez sauver tous les fichiers du RAM Disque en utilisant la commande COPY *.*, ou en définissant la liste des fichiers à sauver dans un fichier .BAT.

```
COPY C:PRG1.EXE D:
COPY C:*.DTA D:
COPY C:TEST.TXT D:
```

En fait, il est même possible d'écrire un fichier .BAT stockant le contenu du RAM Disque sur un disque PC, vidant le RAM Disque, puis le remplissant avec une série de fichiers PC.

```
COPY C:*.* D:
DEL C:*.*
COPY D:UTIL.EXE
COPY D:*.DTA
```

On peut alors disposer de plusieurs configurations de travail différentes. Par exemple, une configuration bureautique, programmation et jeux.

Quelques points faibles

Le Portfolio ne gère pas les disques durs. De ce fait Trans-Drive non plus, ce qui implique la copie des fichiers sur une disquette ou un RAM Disque avant de les transférer. Le Portfolio utilise *MSDOS 2.11* parce qu'il ne prend pas beaucoup de place en mémoire et convient donc parfaitement pour un ordinateur ne disposant, en standard, que de 128 Ko de mémoire.

Trans-Drive ne reconnaît que les disquettes 5 pouces 1/4 1,2 mégas et les disquettes 3 pouces 1/2 1,44 mégas. Si vous ne possédez qu'un PC XT avec des lecteurs 5 pouces 1/4 360 Ko ou 3 pouces 1/2 720 Ko, il faudra créer des disquettes de transfert en les formatant avec un utilitaire spécial, fourni avec les disquettes de Trans-Drive.

Pour finir

Trans-drive est un produit performant qui pourra vous rendre de nombreux services, surtout si vous manipulez beaucoup de fichiers. C'est, par exemple, un excellent support pour un développeur qui désire écrire des programmes avec les outils PC, pour pouvoir ensuite les transférer sur Portfolio.

Patrick Leclercq ■

Edité par **Becker & Partner**
Distribué par **Lexiel**
Pour Portfolio
Prix: 545 F

PCCOMMAND
Utilisez votre ST comme un PC

Découvrez, grâce à un logiciel du domaine public, les possibilités d'un système d'exploitation type PC sur votre ST.

Le ST dispose d'une interface graphique simplifiant la manipulation des fichiers. En revanche, le PC ne dispose en standard que d'une interface de commandes du type interpréteur de ligne, où il faut taper manuellement des commandes. L'utilisation d'une interface graphique facilite l'emploi de l'ordinateur aux débutants, mais un interpréteur de commandes possède aussi des avantages.

Présentation de PCComand

Le logiciel *PCComand* est un interpréteur de commandes ST conçu pour ressembler à l'interpréteur de commandes des compatibles PC. Il peut permettre à un utilisateur débutant de s'initier à la philosophie d'utilisation des PC. Il s'utilise avec des mots-clés. Toutes les opérations courantes réalisées avec *GEM* (affichage d'informations, effacement, copie, changement de nom, etc.) peuvent être faites avec *PCComand*. La commande DIR permet d'afficher les fichiers d'un disque. La commande DEL permet d'effacer un fichier. La commande COPY permet de copier un fichier d'un disque à un autre, ou d'un répertoire à un autre. La commande RENAME, ou REN permet de changer le nom d'un fichier.

Paramétrage des commandes

L'une des supériorités de l'interpréteur de commandes par rapport à l'interface graphique vient du fait qu'il est possible de paramétrer les commandes. Par exemple, pour effacer tous les fichiers d'un disque ou d'un répertoire, il suffit de taper DEL *.*. Le symbole *.* indique à l'interpréteur de commandes que tous les fichiers doivent être effacés. La commande DEL *.GFA n'efface que les fichiers ayant une extension .GFA. De même, la commande DEL TEST.* n'efface que les fichiers TEST, quelque soit leur extension. Pour réaliser la même chose avec *GEM*, il faut sélectionner les fichiers un à un, puis déplacer le tout sur le symbole de la poubelle. De la même manière, il est simple d'écrire COPY A:*.GFA B: pour copier tous les fichiers GFA du disque A sur le disque B. Alors que la même opération est nettement moins pratique à réaliser avec *GEM*.

Liste des commandes de PCComand

PCComand possède de nombreuses commandes dont certaines sont assez hermétiques. Voici les plus courantes:

BELL: Emet un beep sonore
BYE: Quitte *PCCOMAND* et revient sous *GEM*
CLS: Efface l'écran
COPY: Copie de fichiers ou de répertoires
DATE: Permet de lire et de modifier la date de l'horloge ST
DEL: Effacement fichiers
DIR: Affichage de fichiers
HELP: Affiche la liste des commandes de PCComand
PRINT: Impression d'un fichier
REN: Changement du nom d'un fichier
TIME: Permet de lire ou de modifier l'heure de l'horloge ST
TYPE: Affiche sur l'écran le contenu ASCII d'un fichier
VER: Affiche le numéro de version du TOS
VOL: Affiche le nom d'un disque

Exécution d'un programme

Pour exécuter un programme *GEM*, il faut taper son nom. La présence de l'extension n'est pas obligatoire.
>DEMO.PRG
Lorsque *PCCOMAND* exécute un programme, il considère que c'est un programme

```
Solid Applications, Inc.
1333 Moon Drive, Yardley, PA 19067
Copyright (c) 1986
All rights reserved
Shareware: Type SHARE for more information.

PCommand V1.00  SAI 05-03-86

E>dir
.              <DIR>     00-00-2028  12:01am  10
..             <DIR>     00-00-2028  12:00am  10
PCCOMAND.TXT     14052   02-09-1992  11:35pm  20
PCCOMAND.BAK     16480   00-00-2028  12:36am  20
PCCOMAND.ASC     14052   00-00-2028  12:02am  20
PCCOM   .PRG     38400   04-24-1986  12:20am  20
PCCOMAND.LIB     16478   00-00-2028  12:39am  20
       7 File(s)

E>
```

Exemple d'utilisation de la commande DIR. Le nom des fichiers est affiché avec la taille mémoire, la date eet l'heure de création, ainsi que le type de fichier (20 pour les fichiers ordinaires et 10 pour les répertoires).

GEM et appelle la souris. La plupart des programmes *TOS* (extension .TOS et .PRG) ne sont pas gênés par la souris, contrairement à d'autres. Ces derniers doivent être appelés par la commande TOS qui supprime la souris.

```
>TOS DEMOGRAF
```

Si le fichier est de type .TTP, il nécessite des paramètres pour s'exécuter, comme par exemple un nom de fichier. Ces derniers peuvent être transmis dans la ligne de commandes. L'exemple suivant transmet la chaîne de caractères "abcdef" au programme PRGTEST.

```
>PRGTEST abcdef
```

La touche [F3] permet de rappeler la dernière commande utilisée. Cela facilite l'utilisation des commandes répétitives.

Gestion des répertoires

GEM gère les manipulations des répertoires de manière simple. Pour entrer dans un répertoire, il suffit de cliquer sur son nom. pour le quitter, il faut cliquer sur le bouton en haut à gauche de la fenêtre d'affichage du répertoire.

La création d'un répertoire se fait avec une option du menu déroulant. Pour effacer un fichier, il faut le mettre à la poubelle.

PCCOMAND gère les répertoires d'une manière très différente. Pour entrer dans le répertoire rep, il faut utiliser la commande CD rep. La commande CD .. permet de sortir du répertoire courant. MD rep permet de créer un répertoire de nom rep. RD permet d'effacer un répertoire.

CD rep Passage dans le répertoire rep
CD .. Sortie d'un répertoire
RD rep Effacement du répertoire rep
MD rep Création du répertoire rep

Les fichiers batchs

L'interpréteur de commandes exécute fidèlement les ordres tapés au clavier, mais il peut aussi exécuter une série de commandes écrites dans un fichier texte. Ce fichier est appelé fichier batch.

Les fichiers batchs doivent obligatoirement avoir l'extension .BAT. Pour exécuter un batch sous *PCComand*, il suffit de taper son nom. Toutes les commandes de *PCComand* peuvent être utilisés dans un fichier BAT (autre nom des fichiers batch).

Voici un exemple d'utilisation des fichiers batchs. Pour imprimer un fichier sous *GEM*, il faut cliquer dessus et choisir l'option IMPRIMER. L'impression de plusieurs fichiers est une opération assez contraignante, puisqu'il faut surveiller l'exécution du travail et relancer le processus pour chaque fichier. Avec *PCCOMAND*, l'impression d'un fichier se fait avec la commande PRINT.

```
>PRINT DOC.TXT
```

En créant un fichier batch, il est possible d'automatiser l'impression de plusieurs fichiers.

Le fichier IMPRIM.BAT imprime à la suite les fichiers DOC.TXT, EXEMPLE.TXT, INFO.TXT et HELP.TXT. La commande REM permet d'intégrer des commentaires dans les fichiers batchs.

```
REM ************************
REM *  Fichier IMPRIM.BAT  *
REM ************************
PRINT DOC.TXT
PRINT EXEMPLE.TXT
PRINT INFO.TXT
PRINT HELP.TXT
```

Pour lancer l'impression, il suffit de taper le nom du fichier BAT. Cela vous donne le temps d'aller boire une tasse de café.

```
>IMPRIM
```

```
E>dir
.                <DIR>     00-00-2020  12:01am  10
..               <DIR>     00-00-2020  12:00am  10
PCCOMAND.TXT       14052   02-09-1992  11:35pm  20
PCCOMAND.BAK       16400   00-00-2020  12:36am  20
PCCOMAND.ASC       14052   00-00-2020  12:02am  20
PCCOM   .PRG       38400   04-24-1986  12:20am  20
PCCOMAND.LIB       16470   00-00-2020  12:39am  20
PCCOM1  .PI3       32066   00-00-2020  12:03am  20
       8 File(s)

E>dir *.txt
PCCOMAND.TXT       14052   02-09-1992  11:35pm  20
       1 File(s)

E>dir *.prg
PCCOM   .PRG       38400   04-24-1986  12:20am  20
       1 File(s)

E>dir *.lib
PCCOMAND.LIB       16470   00-00-2020  12:39am  20
       1 File(s)

E>
```

Exemple d'utilisation de la commande DIR avec paramètre précisant le type de fichier à afficher. Visualisation des fichiers *.TXT, *.PRG et *.LIB.

Création de fichiers batchs

Les fichiers BAT peuvent être créés avec un éditeur de texte quelconque, ou avec la commande QED. Celle-ci appelle un mini-éditeur de texte intégré dans *PCCOMAND*. Les fonctions de QED sont malheureusement très limitées.

Il ne peut être comparé à un éditeur de texte classique, mais suffit pour créer des fichiers BAT.

```
>QED test.bat
```

Création d'un fichier BAT avec la commande QEB. Le contenu du fichier est ensuite visualisé avec la commande TYPE.

```
>qed test.bat
Enter a line with no characters to end input.

: copy *.gfa a:
: copy *.pil a:
: copy infos.dta a:
: copy lisezmoi.txt a:
:
Saving file.

>type test.bat
copy *.gfa a:
copy *.pil a:
copy infos.dta a:
copy lisezmoi.txt a:

>dir *.bat
TEST    .BAT          71  00-00-2020  12:02am  20
       1 File(s)

>
```

Sauvegarde partiel de disque dur

Voici maintenant un autre exemple d'utilisation des fichiers batchs qui peut servir aux utilisateurs de disque dur. Un disque dur est un outil très puissant qui rend de nombreux services, mais qui peut provoquer une catastrophe en cas de problème matériel ou de virus. En théorie, il faut sauver fréquemment les données sur disque, mais l'expérience prouve que très peu d'utilisateurs le font. Cela est du notamment au temps perdu à effectuer la sauvegarde. Il est possible de faire une sauvegarde globale de tous les fichiers en copiant la totalité d'un répertoire ou d'un disque, mais cela oblige à sauver tous les fichiers présents. Il est préférable de ne sauver que les fichiers de travail, afin de gagner du temps et de la place mémoire. En utilisant un fichier batch, il est possible de définir avec exactitude les fichiers à sauver. Le fichier batch suivant sauve automatiquement tous les fichiers GFA et PI1 présents dans un répertoire, ainsi que les fichiers LIVRES.DTA et INFOS.TXT.

```
REM ***************************
REM *  Fichier de sauvegarde  *
REM ***************************
```

```
COPY *.GFA A:
COPY *.PI1 A:
COPY LIVRES.DTA
COPY INFOS.TXT
```

En sauvant régulièrement votre travail, vous prenez une bonne assurance contre les accidents de disque dur. Cette technique peut également être mise en œuvre par les utilisateurs de RAM disque.

Passage de paramètres aux fichiers batchs

Il est possible de passer des paramètres aux fichiers batch. Cela permet d'augmenter la souplesse d'utilisation des batchs. Supposons, par exemple, que vous deviez faire subir plusieurs traitements à un fichier, comme une compilation et une impression. Le fichier EXEC.BAT réalise une compilation et une impression sur le même fichier.

>EXEC fic.c

Lors de l'appel de EXEC.BAT, l'interpréteur de commande remplace le symbole %1 par le nom du paramètre de EXEC. Avec plusieurs paramètres, le premier est traité dans le fichier BAT sous le nom de %1, le second sous le nom de %2, etc..

```
REM **************************
REM *  Fichier EXEC.BAT       *
REM **************************
COMPIL %1
PRINT %1
```

Il est possible de réaliser des fichiers BAT complexes utilisant plusieurs paramètres. Cela ouvre la voie à de nombreuses possibilités.

Utilisation de PCComand avec le Sozobon C

Les cours d'initiation au *C* d'*Atari Magazine* utilisent le *Sozobon C*. Ce dernier est très performant pour un logiciel du domaine public, mais son utilisation est parfois comple xe. La compilation se fait avec le programme CCD.TTP qui génère un programme .TTP à partir d'un source C. L'exemple suivant génère le fichier TEST.TTP à partir du fichier TEST.C >CCD test.c

Cette syntaxe n'affiche pas l'état de la compilation. C'est assez désagréable de voir un écran vide sans savoir ce qui se passe. L'option -V indique au *Sozobon C* d'afficher les différentes étapes de compilation.

>CCD test.c -V

Les fichiers .TTP commencent par afficher une boîte de dialogue permettant à l'utilisateur de communiquer des informations aux programmes. C'est gênant si les programmes n'ont pas besoin d'informations complémentaires. Pour changer cela, il faut transformer le nom du fichier en TEST.TOS, ou utiliser l'option -O du *Sozobon C* pour produire un fichier d'un nom précis. L'exemple suivant génère le fichier TEST.TOS à partir du fichier TEST.C.

>CCD test.c -O test.prg -V

Afin d'éviter de taper tout le temps la même séquence, il est possible d'utiliser un fichier BAT. Le fichier CC.BAT génère un fichier PRG à partir d'un fichier C.

```
REM ********************
REM *  FICHIER CC.BAT  *
REM ********************
CCD %1.C -O %1.PRG -V
```

Pour utiliser ce fichier BAT, il faut lui donner le nom du fichier à compiler. Ce nom ne doit pas comprendre l'extension .C afin de simplifier l'écriture du fichier batch.

>CC test

Les récents articles sur le C traitent de la programmation graphique. Ils utilisent la bibliothèque graphique *GEMFAST* qui est assez puissante, mais qui nécessite une syntaxe de compilation complexe. Pour compiler le source GRAPH.C en programme GRAPH.PRG utilisant les fonctions de *GEMFAST*, il faut linker le programme avec les bibliothèques AESFAST.A et VDIFAST.A. Cela se fait de la manière suivante: «>CCD graph.c -O graph.prg -V AESFAST.A VDIFAST.A» Chaque compilation n'est pas très simple à taper! Le fichier CCDGEM.BAT simplifie la compilation des programmes C utilisant la bibliothèque graphique *GEMFAST*.

```
REM ************************
REM *  Fichier CCDGEM.BAT  *
REM ************************
CCD %1.C -O %1.PRG -V AESFAST.A VDIFAST.A
```

Pour utiliser ce fichier, il suffit de lui communiquer le nom du fichier C à compiler.

>CCDGEM graph

Fonctions batchs évoluées

Les fichiers batchs sont capables de faire des choses complexes, pratiquement comme un langage de programmation. Ils sont dotés de commandes permettant de réaliser des tests, de vérifier si un fichier ou un répertoire existe, de faire des branchements conditionnels, d'afficher des messages et de demander des informations à l'utilisateur. Cela permet de créer des applications relativement performantes. La commande REM insère des commentaires dans les fichiers batchs. Bien qu'au niveau fonctionnelle, elle ne serve à rien, elle est importante pour écrire des programmes lisibles et documentés. La commande EXIST teste si un fichier ou un répertoire existe. Elle peut, par exemple, vérifier si le fichier sur lequel veut agir l'utilisateur existe vraiment. Cela évite d'appeler un programme avec comme paramètre le nom d'un fichier inexistant. La commande INPUT saisit une chaîne de caractères. Elle permet de demander la confirmation d'une opération avec un message du type «Voulez-vous vraiment copier ce fichier Oui/Non?». Dans un programme d'installation, elle saisit le nom d'un répertoire, ou de préciser la manière dont l'installation doit se faire "1) Installation partielle; 2) Installation complète».

La commande GOTO permet de brancher directement l'exécution du fichier BAT à un label. Les labels sont des mots commençant par le symbole «:». La commande IF teste la valeur d'une expression. Elle permet, par exemple, d'exécuter une série de commandes en fonction du type de TOS, ou d'une information saisie par l'utilisateur. Ces commandes sont les plus importantes. Il en existe d'autres qui correspondent à des cas particuliers, comme le traitement des erreurs d'exécution.

Où trouver PCComand?

Il est disponible en téléchargement sur le serveur **3615 ATARI** sous le nom de PCCOMAND.ARC. Sa taille est de 32343 octets. Il se décompacte avec le logiciel ARC. Il est accompagné d'une documentation anglaise détaillant ses fonctionnalités et la création des fichiers batchs.

Pour finir

Vous connaissez maintenant le fonctionnement de l'interpréteur de commandes *PC-Comand*. Ce programme vous permettra de réaliser des choses intéressantes sur ST.

A présent, vous en savez beaucoup plus sur la philosophie d'utilisation des compatibles PC.

Patrick Leclercq ■

BMS

(16) 89.60.30.27

Les Applications de l'Informatique

7, rue Schlumberger
68200 MULHOUSE
Fax: 89 42 52 85



SCANMAN 256



L'ensemble comprend un scanner Logitech 256 niveaux de gris, résolution jusqu'à 400 DPI, une interface port cartouche avec son alimentation et le logiciel de traitement d'image REPRO STUDIO Junior.



SCANMAN PLUS



L'ensemble comprend un scanner Logitech 32 niveaux de gris, résolution jusqu'à 400 DPI, une interface port cartouche avec son alimentation et le logiciel de traitement d'image REPRO STUDIO Junior.



REPRO STUDIO JUNIOR
Version universelle



Logiciel de traitement d'image, double passage largeur. Gère les scanners Printtechnik, Cameron 2, Epson, A- Magic, Genius et Import fichiers PC3, PI3, IMG, PAC, TIF... Dessin main levée, éditeur de motifs, fonctions blocs, masquage d'une image, filtres, contraste, luminosité.



AVANT VEKTOR



AVANT VEKTOR est un logiciel de dessin vectoriel, intégrant un module de conversion d'image bitmap en vecteurs. Conversion automatique ou manuelle, zoom, rotation, miroir, import/export CVG, GEM, EPS. Superposition de l'image bitmap et de l'image vecteur.

AVANT VEKTOR PLOT



Vectorisation, dessin vectoriel, import/export CVG, GEM, EPS. Editeur de texte avec fontes CFN (CALAMUS®). Module de découpe en mode HPGL et GPGL.
Ce logiciel est particulièrement adapté aux **découpes de lettres** et des dessins sur supports adhésifs ou masques de sérigraphie.



AVANT TRACE

990F

Version de la gamme AVANT VEKTOR comprenant uniquement le module de vectorisation.



VERSION PC, MAC et ATARI



Scanner couleur A4, 300 DPI, 16.7 Millions de couleurs ou 256 niveaux de gris, interfacé SCSI pour MAC, PC et ATARI TT. Fonction Zoom de 12.5 % à 800 %.
Temps de scan : 9.1 s pour 256 niveaux de gris, 90s en mode couleur 24 bits.
Avec Colorscan, vos documents, même en flashage, seront restitués avec la plus grande fidélité.

Bon de commande

A retourner : BMS 7, rue Schlumberger 68200 MULHOUSE Fax (16) 89 42 52 85

	Prix TTC	Quantité	TOTAL
Scanman 256	3690 F		
Scanman plus	2390 F		
Repro Studio Junior	890 F		
Avant Vektor	2890 F		
Avant Vektor Plot	5490 F		
Avant Trace	990 F		
Colorscan	15900 F		

MONTANT TOTAL ____
PORT 40.00 F
NET A PAYER ____

REGLEMENT PAR
☐ Chèque à la Commande
☐ Contre Remboursement

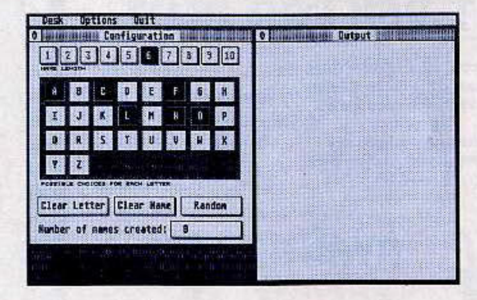
Signature

LES NOUVEAUTES DU 3615 ATARI
La détente à l'honneur

Le domaine public sur ST est toujours aussi dynamique. Voici encore une bonne vingtaine de nouveautés, sélectionnées pour vous par le 3615 ATARI et disponibles en téléchargement...

Alpha.TOS

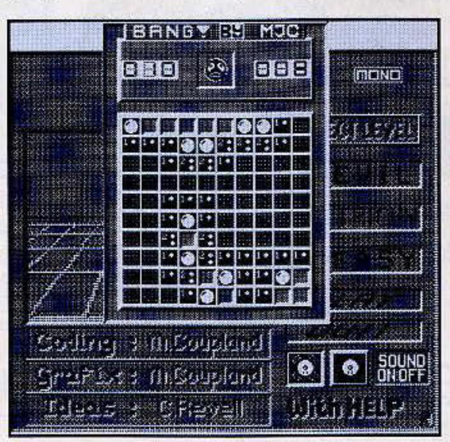
Vous cherchez un prénom à votre enfant? Vous êtes en train d'écrire un roman et vous peinez sur le nom des personnages? Voici *Alpha*! Un programme qui génère automatiquement, à partir des lettres que vous lui



définissez, toutes les combinaisons de mots possibles. C'est entièrement sous GEM et très convivial. Plein d'applications sont possibles... Toutes résolutions. Toutes machines.

Bang_COL.TOS

Voici un petit jeu de réflexion directement inspiré d'un des plus célèbres jeux disponibles sous *Windows 3* sur PC. Cette version ST y apporte quelques améliorations graphiques et sonores. Le jeu se présente sous



forme d'un quadrillage, dont chaque case est au départ cachée. Certaines de ces cases contiennent une bombe, d'autres des indications sur les cases mitoyennes comportant les bombes. L'objectif et de découvrir toutes les cases, sauf celles renfermant les bombes. C'est un jeu simple mais très prenant.
Fonctionne sur ST, STE, MEGA ST/STE, TT. Couleur.

Bang_HI.TOS

Il s'agit là de la version monochrome du jeu précédent `Bang_COL.TOS`.
Fonctionne sur ST, STE, MEGA ST/STE, TT. Monochrome.

Campaign.TOS

Voici la version de démonstration du futur jeu d'*Empire*. Cette démo vous permet de reconstituer toute un bataille de tanks. Superbe et en 3D faces pleines.
Fonctionne sur STF/STE et MEGA ST/STE. Couleur.

Chicon.TOS

Un fichier RSC plein d'icônes pour votre bureau Newdesk.
Toutes résolutions. MEGA STE et TT.

DC_RT_DC.TOS

Pour les «nulos» du double clic voici un utilitaire qui remplace le double clic par un simple clic sur le bouton droit...
Toutes résolutions. Toutes machines.

DSK.TOS

«Data Statement Kit» s'adresse aux programmeurs *Assembleur* sur *Devpac* ayant l'habitude de mélanger *Assembleur* et *GFA Basic*. *DSK* génère automatiquement les lignes de DATA.
Toutes machines. Couleur.

Entfiles.TOS

Encore un programme de gestion de vos disques, K7, films, livres et autres CD.
Les options sont nombreuses, et même si le logiciel n'est pas sous GEM, il est très simple à utiliser.
ST/STE, MEGA ST/STE. Couleur.

Flipped.TOS

Et hop, encore un petit jeu sympa très bien réalisé (graphismes colorés et sons digitalisés). C'est un jeu de réflexion pour solitaires...
ST/STE, MEGA ST. Couleur.

Funpack1.TOS

Comprend trois jeux fonctionnant en programme ou en accessoire:
Flipflop, *Puzzle* et *Tic_Tac*.
Toutes résolutions. Toutes machines.

FZDT.TOS

Voici un nouveau logiciel de communication pour tous les fans de Compuserve, Genie et autres BBS. Vraiment puissant.
ST/STE, MEGA ST/STE. Couleur et monochrome.

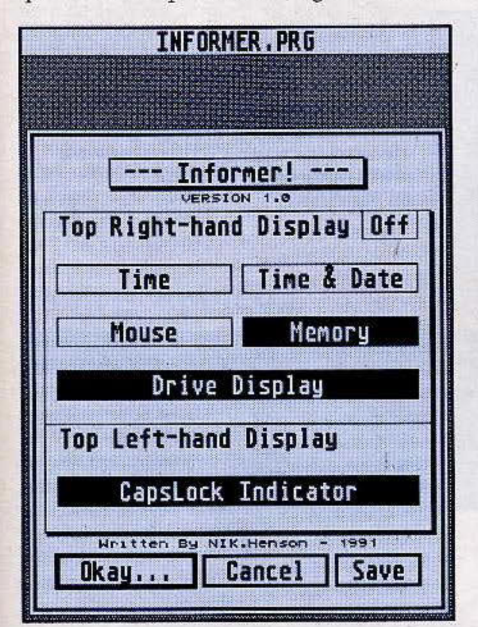
Incontrol.TOS

Voici un logiciel de contrôle des événements MIDI universels. Il utilise un système de sliders original et très convivial. Un logiciel que tous les musiciens, qui regrettent que

leurs synthés numériques ultra modernes n'aient plus les potentiomètres des anciens analogiques, se doivent de posséder.
Fonctionne en programme et en accessoire de bureau.
Toutes résolutions. ST/STE, MEGA ST/STE.

Informer.TOS

Un fantastique utilitaire pour tous ceux qui s'inquiètent de savoir ce qui se passe dans leur système. C'est à la fois un TSR et un accessoire de bureau. Parmi les informations qu'il affiche en permanence, signalons un in-



dicateur de [CapsLock], les coordonnées de la souris, le lecteur en cours d'accès, l'heure, etc.
Toutes résolutions. Toutes machines.

Intro.TOS

Un super utilitaire très convivial pour créer des «intros» animées et sonores ou des disquettes de démonstration.
Toutes machines. Couleur.

Keyshow.TOS

Un accessoire qui permet de contrôler l'attribution des touches de votre clavier.
Toutes résolutions. Toutes machines.

Miditk.TOS

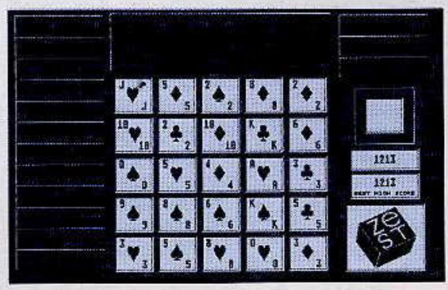
MidiTalk est un utilitaire très pratique pour comprendre ou tester une installation MIDI. Il capte tous les événements MIDI qu'il reçoit, les affiche et les stocke dans un fichier.
Toutes résolutions. Toutes machines.

Miditool.TOS

Deux utilitaires en accessoire pour musiciens MIDI.
Local permet de mettre en mode KLocal On/off un canal MIDI donné.
Panic envoie instantanément un All Notes Off sur tous les canaux MIDI.
Toutes machines. Toutes résolutions.

PLAXATAX.TOS

La plaque dentaire attaque vos gencives et vos dents. Heureusement, *Plaxatax* est là!



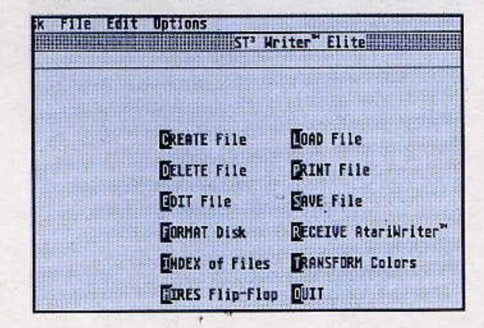
On croirait une mauvaise pub pour un dentifrice, mais *PlaxAtax* est un jeu d'arcade amusant.
ST/STE. Couleur.

POKERSQ.TOS

Pour les fans de Poker et de solitaires voici un étonnant jeu basé sur le Poker où l'on pose les cartes sur un échiquier afin de créer un maximum de combinaisons gagnantes en diagonal, vertical ou à l'horizontal.
Monochrome uniquement.
Toutes machines.

QUIZICAL.TOS

Un Quizz pas si simple pour réviser vos connaissances et votre culture.
Attention, c'est en anglais.
ST/STE. Couleur.



STOS3D.TOS

Une démo des fabuleuses possibilités de STOS 3D sous forme de jeu.
ST/STE. Couleur.

STWEL46.TOS

Vous connaissez tous *ST Writer*, le premier traitement de texte apparu sur ST il y a plus de six ans! Voici la toute dernière mouture: la version 4.6.
Toutes résolutions. Toutes machines.

TURBORAM.TOS

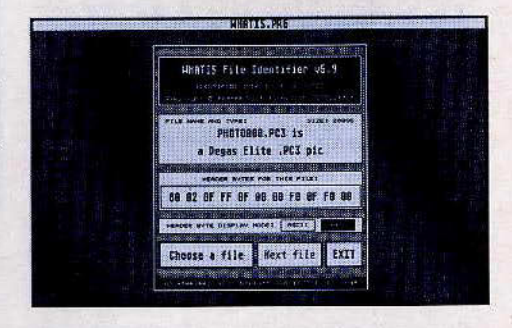
TurboRam IV est probablement le plus rapide et le plus puissant des Ram disques résistants au Reset. Un must.
Toutes résolutions. Toutes machines.

VOLUME

Un logiciel de conception d'objets en 3D que vous pouvez ensuite animer. Très réussi.
Toutes machines. Couleur uniquement.

WHATIS59.TOS

La nouvelle version 5.9 de *Whatis*, le logiciel accessoire de bureau, qui reconnaît plus de 125 formats de fichiers!
Toutes résolutions. Toutes machines.



POUR LE PORTFOLIO

DSPPGC.ZIP

Logiciel ST permettant de visualiser les images au format Portfolio PGC sur ST et TT.
Toutes résolutions. Toutes machines.

SECURE.ZIP

Voici un petit logiciel affichant une animation lors du démarrage du Portfolio.

STHNGE.ZIP

Une animation basée sur le célèbre site Stonhnge. ■

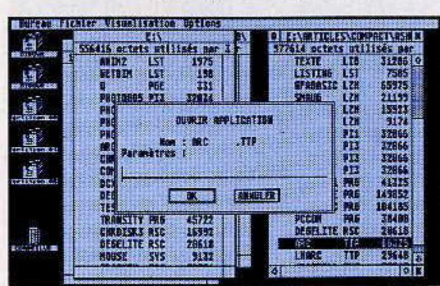
LE THEME DU MOIS
Les compresseurs de données

La place mémoire est souvent limitée, surtout avec des disquettes. Les compresseurs de données permettent de réduire efficacement la taille des fichiers et des programmes.

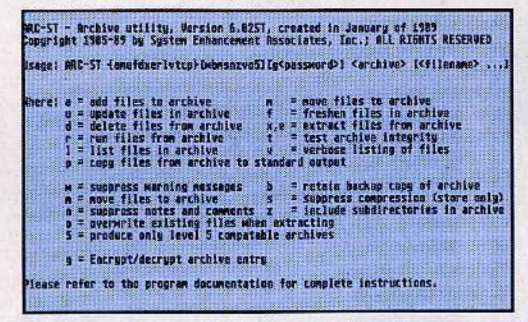
Il existe différentes méthodes de compression de fichiers. La technique la plus courante est de coder les séquences d'octets répétitives, comme, par exemple, les zones de noir dans une image. Le format de compression le plus répandu sur ST est le format .ARC. Il est sérieusement concurrencé par le format .LZH, plus récent et plus performant. Plusieurs logiciels du domaine public permettent d'utiliser ces formats de compression.
Ils analysent les données et déterminent la meilleure façon de compactage avant de commencer la phase de compression. Les fichiers contenants des données compressées sont généralement appelés des fichiers archives, ou tout simplement des archives.

Le programme ARC.TTP

Ce logiciel, grand classique de la compression de fichiers, nous vient du monde PC.
Il compresse ou décompresse des fichiers au format .ARC et en manipule aussi le contenu. La version 6.02 est nettement plus performante que les précédentes. Le fichier compacté, contenant le programme .TTP et sa documentation, se nomme ARC602.ARC. Son contenu peut être décompacté avec n'importe quelle autre version de *ARC*, ou avec ARCX.TTP (voir plus loin).



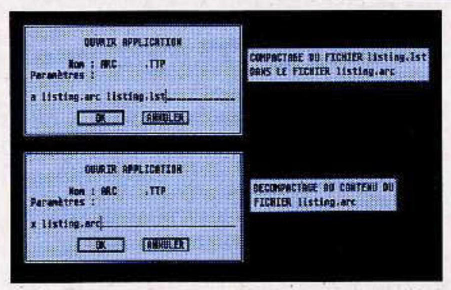
Le programme ARC est prêt à recevoir des ordres. La fenêtre de dialogue TOS est ouverte sur le bureau.



La liste des commandes de ARC. Cette liste est obtenue par défaut lorsque l'utilisateur ne donne aucun ordre au programme.



Compilation de plusieurs fichiers dans le même fichier .ARC. On peut éventuellement compiler un grand nombre de fichiers dans la même archive .ARC.



Exemples d'utilisations de ARC. La première fenêtre montre comment compiler un fichier. La seconde est un exemple de décompilation.

Pour utiliser ce programme, il suffit de cliquer dessus et taper une ligne de commandes correspondant à la tâche voulue. Par exemple, pour compresser le contenu du fichier TEST.TXT dans le fichier TEST.ARC, il faut taper la séquence:
«a TEST.ARC TEST.TXT».
La commande «a» correspond à la compression. Si le fichier TEST.ARC n'existe pas, il est créé. En revanche, s'il existe déjà, le fichier TEST.TXT vient s'ajouter au contenu du fichier .ARC. Il est donc possible de stocker plusieurs documents dans le même fichier .ARC.
Voici la liste des commandes les plus courantes:

- a = ajouter des fichiers dans un fichier .ARC
- u = mettre à jour des fichiers dans une archive
- d = détruire un fichier dans une archive
- l = afficher la liste des fichiers d'une archive
- p = copier un fichier d'une archive vers la sortie courante (imprimante en général)
- x,e = décompactage d'un ou plusieurs fichiers
- t = test si la structure d'une archive n'est pas altérée
- v = affiche la liste des fichiers avec tous les détails (taille initiale, taille compressée, taux de compression, méthode de compression, date de compression et valeur CRC permettant de contrôler que le

contenu du fichier n'est pas altéré).
g = codage du contenu d'un fichier

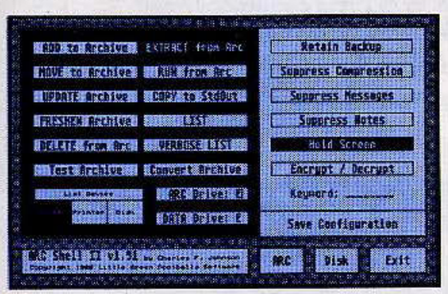
La commande g permet de crypter le contenu d'un fichier avec un code quelconque. C'est une fonction importante, qui assure une certaine sécurité informatique. Dans plusieurs cas, il est plus prudent de crypter les fichiers devant être envoyés par la poste ou téléchargés.

Le programme ARCX.TTP

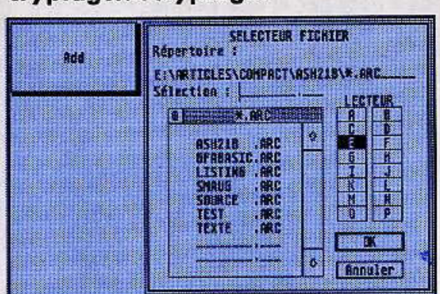
Ce petit programme est juste un décompacteur de fichiers .ARC. Il ne prend pas beaucoup de place et rend bien des services. Il est présent sur la disquette de téléchargement d'Atari Magazine et peut être téléchargé à partir du 3615 ATARI. Son emploi est des plus simples, puisqu'il suffit de cliquer dessus et d'indiquer le nom du fichier .ARC à décompacter.

Shell graphique pour ARC

Le programme *ARCSHELL* est un shell graphique pour *ARC*. C'est en fait une interface GEM qui sert d'intermédiaire entre l'utilisateur et le programme ARC.TTP. Elle donne la possibilité d'utiliser toutes les fonctions de *ARC* en cliquant avec la souris. Le choix des fichiers à compacter ou à décompacter se fait avec des sélecteurs d'objets. La version 1.91 de ce programme est disponible en téléchargement sous le nom de *ARC.PRG*. Il est indispensable de posséder cette interface si vous avez l'intention d'utiliser souvent le programme *ARC*.



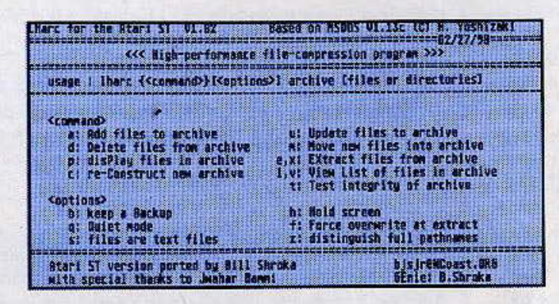
Le menu de ARCSHELL version 1.91. Toutes les fonctions du programme ARC sont disponibles. Il suffit de cliquer sur la bonne case. L'ergonomie de ce programme est excellente. Vous pouvez remarquer, à droite de l'écran, la présence de la fonction de codage de fichiers. La zone de saisie des codes de cryptage est grisée. Elle devient accessible après avoir cliqué sur la case cryptage/décryptage.



ARCSHELL: Le choix des fichiers à manipuler se fait avec un sélecteur d'objets.

Le programme LHARC.TTP (LZH)

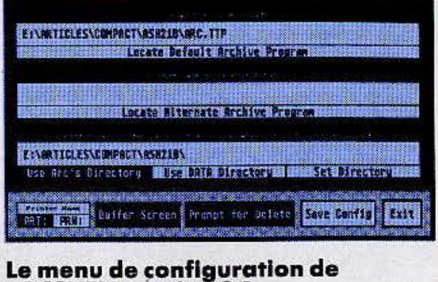
Ce logiciel est une version très améliorée de *ARC.TTP*. Il fonctionne de la même manière, mais utilise le format de fichier .LZH qui est plus compact que le format .ARC.



La liste des commandes de LHARC version 1.02. De la même manière qu'avec le programme ARC, la liste des commandes est affichée lorsque l'utilisateur ne donne pas d'ordres précis à LHARC.

Deux versions de *LHARC* sont disponibles sur le 3615 ATARI. La version 1.02 est stockée dans le fichier LHARC2.ARC.
La version 2.01 est stockée dans le fichier LZH201.TOS. Ce dernier est un programme auto-décompactable.
La version 2.01 est plus rapide et plus performante que la version 1.02, mais elle ne fonctionne pas avec l'interface graphique *ARCSHELL*.

Shell graphique version 2



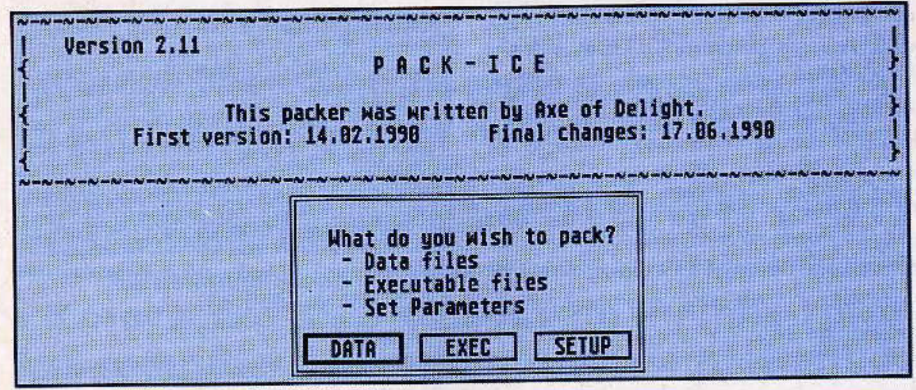
Le menu de configuration de ARCSHELL version 2.1.

Comme nous l'avons vu, *ARCSHELL* est une interface graphique qui permet de les utiliser avec une souris et des sélecteurs d'objets. C'est un programme qui simplifie la vie de l'utilisateur.
Le fichier ASH21B.ARC contient la version 2.1 de *ARCSHELL*. Celle-ci se nomme *ARCSHL21.PRG*. Attention: ce logiciel n'est pas un programme complet, mais juste une interface entre l'utilisateur et les programmes de compression.
ARC.TTP et LHARC.TTP doivent se trouver dans le même répertoire que ARCSHL21.PRG.
Toutes les fonctions peuvent être appelées en cliquant sur des zones de l'écran. Des sélecteurs d'objets facilite le choix des fichiers à manipuler. Une option située en bas de l'écran précise si les manipulations doivent se faire sur des fichiers .ARC ou .LZH.
Attention: La version 2.01 de *LHARC* ne fonctionne pas avec cette interface. Vous devez donc utiliser uniquement *LHARC* version 1.02.

Le menu de ARCSHELL version 2.1. Ce menu ressemble à celui de la version 1.91, mais il possède quelques fonctions en plus. Notamment l'utilisation des fichiers de type .LZH. Le sélecteur .ARC/.LZH est situé en bas à gauche de l'écran.

Le programme PFXPAKE

Le programme *PFXPAKE* est un compacteur de programmes, c'est-à-dire qu'il réduit la taille des fichiers exécutables (.PRG, .TTP et .TOS). Les fichiers compactés restent exécutables. En fait, cet utilitaire compacte les programmes sous forme de données qui sont décompactées en mémoire. Le début du fichier compacté est une petite routine qui décompacte le logiciel directement dans la mémoire de l'ordinateur.
Afin de tester ce produit, nous avons com

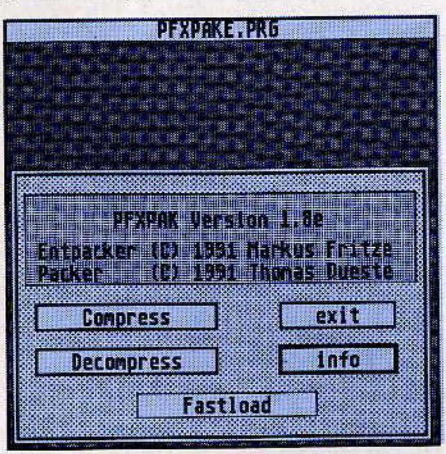


Le menu utilisateur de PACK-ICE.

pacté le *GFA Basic 3.5E*. La taille de ce dernier est passée de 104185 octets à 77892 octets. Cela représente un gain de place de 25%, ce qui n'est pas négligeable et peut rendre bien des services à un utilisateur de disquettes. Pour nous divertir, nous avons compacté *PFXPAKE* avec lui-même. Il en résulte une nouvelle version de *PFXPAKE* qui fonctionne parfaitement, mais ne fait plus que 6518 octets à la place de 8618 octets (gain de 24.4%). Le résultat a été nettement plus impressionnant avec le logiciel *Degas Elite*, puisque sa taille est passée de 149852 octets à 63970 octets, soit un taux de compactage de 57.3%. Cette différence dans les taux de compression vient probablement du fait que *Degas Elite* a été écrit en langage C, alors que les autres programmes ont été écrits en Assembleur.

Après avoir été compactés, les programmes ARC.TTP version 6.02 et ARCSHL21.PRG ont réduit de moitié.

Une fois packés, les programmes se chargent plus rapidement en mémoire. Selon les programmes, le gain de temps oscille entre 20% et 40%. Cette différence de vitesse vient du fait que la décompilation est plus rapide que la vitesse de lecture du lecteur de disquette. La version non compactée de *Degas Elite* se charge en 22 secondes, alors que la version packée se charge en 14 secondes. Sur un disque dur, on ne note pas de différence lors du temps de chargement. Il n'est pas absolument certain que ce compacteur fonctionne sur tous les programmes, mais tous les utilisateurs de disquettes devraient l'avoir. Téléchargez le fichier LZH201.TOS qui contient la version 2.01 de *LHARC*, ainsi que PFXPAKE.PRG.



Le menu utilisateur de PFXPAKE. Vous remarquerez la présence d'une option de décompression qui redonne aux programmes compressés leur taille initiale.

Le programme PACK-ICE

Il existe un autre compacteur de programmes sur le 3615 ATARI: *PACK-ICE*. Il compresse légèrement mieux les fichiers que *PFXPAKE*, mais travaille très lentement. La compression d'un programme d'une centaine de kilo-octets peut dépasser les 15 minutes, alors que *PFXPAKE* le fait en quelques dizaines de secondes. Il a aussi le désavantage de donner au programme compacté le même nom que le programme d'origine, détruisant ainsi ce dernier. De plus, certains logiciels supportent mal le compactage de *PACK-ICE*. Par exemple, une fois compacté, le *GFA Basic* affiche trois bombes et plante.

Patrick Leclercq

Le format .ARC et le format .LZH

Afin de comparer ces deux formats, nous avons compactés quatre fichiers différents: un listing en GFA Basic, sauvé sous forme ASCII, un texte au format du Rédacteur (.LIB), une image Degas Elite, et un programme exécutable. L'image Degas représente un dragon rouge crachant le feu. Elle est disponible en téléchargement sous le nom de SMAUG.PC1. Le programme exécutable est l'interpréteur GFA Basic version 3.5E.

La tableau suivant compare les performances respectives des formats .ARC et .LHZ (version 1.02).

	Taille initiale	Fichier .ARC	Fichier .LZH
Listing .LST	7 585	2 925 (61.4%)	1 755 (76.8%)
Image Degas	32 066	26 522 (17.3%)	21 715 (32.3%)
Texte .LIB	31 286	13 733 (56.1%)	10 124 (67.6%)
GFA Basic	104 185	87 847 (15.7%)	67 547 (35.2%)

Ces chiffres ne sont pas des constantes, mais des ordres de grandeur. Il est impossible de dire que LZH compresse toujours les images avec un taux de 32% et que ARC réduit toujours les listings de 61%. En revanche, il est possible de dire que les performances des compacteurs ne sont pas à négliger et que LZH est toujours plus performant que ARC.

Cependant, il faut noter que le premier compresse les données beaucoup plus lentement que le second. Dans certains cas, LZH met plusieurs minutes à compresser des fichiers que ARC compresse en une vingtaine de secondes.

Compte tenu de la grande complexité de l'image, on peut considérer que son taux de compression n'est pas énorme. Le taux de compression d'un fichier est inversement proportionnel à sa complexité. Si l'image est stockée sous la forme d'un fichier Degas compressé (.PC1), elle occupe 29 081 octets, ce qui correspond à un taux de compression de 9.3%. Il est intéressant de comparer ce chiffre avec les 32.3% de compression obtenue avec un .LZH!







ADAPTATEUR VIDEO CGA PC –> TELE
Réf. GP 210
(Sortie audio et alimentation 12V fournie) ~~700 F~~ **550 F**

SWITCH LASER
Réf. DS 192A
Connexion de 2 Atari sur une laser Atari (Port DMA) **2 000 F**



COMMUTATEUR VIDÉO MULTISYNC
Entrée/Sortie Audio
DB15 HD 3 résolutions
Réf. GP 250 **300 F**



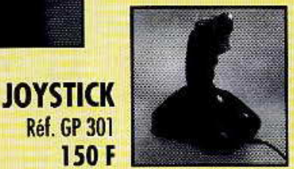
COMMUTATEUR VIDEO Mono/Couleur
Réf. GP 240 **200 F**

SWITCHER de Joystick et de Souris
Réf. GP 220 **150 F**

SOURIS SANS FIL
Livrée avec tapis et rallonge
Réf. GP 303 **550 F**



SOURIS TRANSPARENTE
Réf. GP 300 T - Livrée avec tapis **200 F**
Compatible ATARI & AMIGA



JOYSTICK
Réf. GP 301
150 F

CONNECTEURS

DB09 MALE/FEMELLE	4,00 F
DB15 M/F	5,00 F
DB19 M/F	8,00 F
DB23 M/F	8,00 F
DB25 M/F	6,00 F
DB37 M/F	12,00 F
DIN ATARI 13 M (Vidéo)	20,00 F
DIN ATARI 13 F Ci	20,00 F
DIN ATARI 13 F (Cordon)	30,00 F
DIN ATARI 14 M (Lecteur)	20,00 F
DIN ATARI 14 F Ci	20,00 F
DIN ATARI 14 F (Cordon)	30,00 F
Capot DB 9 Vis longues	6,60 F
Capot DB 19	8,50 F
Capot DB 23	8,50 F
Capot DB 25	6,80 F

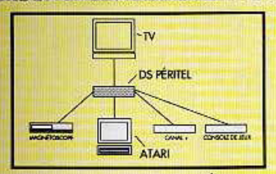
CABLES IMPRIMANTE

1,80 m	Réf. BB 304	48,00 F
3,00 m	Réf. BB 304-3	80,00 F
5,00 m	Réf. BB 304-5	120,00 F
7,00 m	Réf. BB 304-7	130,00 F
9,00 m	Réf. BB 304-9	222,00 F

CABLES SÉRIE 25 M/M ou M/F

	Réf M/M	*Réf M/F*	
1,80 m	BB 305	BB306	48,00 F
3,00 m	BB 305-3	BB306-3	80,00 F
5,00 m	BB 305-5	BB306-5	120,00 F
7,00 m	BB 305-7	BB306-7	130,00 F
10,00 m	BB 305-10	BB306-10	222,00 F

GAMME D'ADAPTATEURS pour Console de Jeux (avec sortie Péritel) pour Moniteurs AMSTRAD, ATARI et COMMODORE **150 F**



PARTAGEUR VIDÉO MANUEL Réf. DS PERI 4
(Péritel 4E/1S) - **486,00 F**
Existe aussi en 2E/1S et 2E/2S

PARTAGEUR D'IMPRIMANTES MANUEL
Réf. DS 252 (2E/1S)
~~210 F~~ **125 F**
Réf. DS 25X (2E/2S)
~~350 F~~ **225 F**
Réf. DS 254 (4E/1S) ~~350 F~~ **225F**



Câble ST/TV péritel ATARI	Réf. GP 901	90 F
Câble HARD COPY	Réf. GP 280	120 F
Détecteur de sonnerie	Réf. GP 302	120 F
Câble minitel ATARI	Réf. GP 225	95 F
Câble Disque Dur ATARI (DMA) 0,80 m	Réf. GP 226	120 F
DB 19 M/DB 19 M 2,00 m	Réf. GP 227	150 F
Câble pour connecter lecteur PC 5" 1/4 sur ST	Réf. GP 242	130 F
Câble pour connecter lecteur PC 3" 1/2 sur ST	Réf. GP 241	130 F
Rallonge joystick 1,80 m	Réf. GP 244	70 F
Rallonge joystick 0,30 m	Réf. GP 243	40 F
Rallonge moniteur DIN 13 M/13 F 2 m	Réf. GP 248	130 F
Rallonge lecteur DIN 14 M/14 F 2 m	Réf. GP 247	130 F
Câble Null Modem (2m) PC –> PC ou PC –> ATARI		
ou ATARI –> ATARI	Réf. AA 332B	150 F
ou ATARI –> MAC	Réf. KOM MAC	140 F
Câble MIDI Atari 1 m	Réf. GP 224	22 F
3 m	Réf. GP 229	45 F
5 m	Réf. GP 230	60 F
Quadrupleur de Joystick ATARI & AMIGA	Réf. GP 246	90 F
Doubleur de Joystick AMSTRAD	Réf. GP 245	60 F

MÉMOIRES

4164 - 10	19 F
41256 - 10	19 F
41464 - 8	32 F
41 1000	65 F
43256 - 8	75 F
44256 - 8	64 F
6264	45 F

CABLES RONDS AU METRE

14 Blindé	30,00 F
Câble en nappe au mètre 50 Conducteurs	20,00 F

SÉLECTEUR DE LECTEURS EXTERNES
Réf. GP 251 **300,00 F**

BARRETTES SIMM
Les 2 Mo x 8 **700 F**

ADAPTATEUR SIMM/SIP
Réf. GP 906 **30 F**

KOMELEC c'est aussi : toute la connectique pour PC, APPLE et AMIGA...

CONCEPTION ET FABRICATION DE CONNECTIQUE SPECIFIQUE SUR DEMANDE

KOMELEC PARIS « Grand Public » *
4, rue Yves Toudic - 75010 Paris - M° REPUBLIQUE
Tél. : (1) 42 08 63 10 / (1) 42 08 54 07
Fax : (1) 42 08 59 05

KOMELEC LYON « Grand Public » *
36, rue Juliette RÉCAMIER - 69006 LYON
Tél. : 78 24 90 60 - Fax : 78 24 76 60

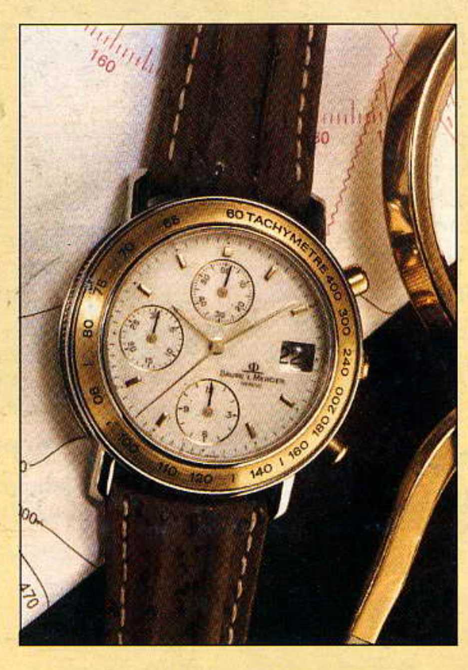
* Ouvert du lundi au samedi de 10h à 12h30 et de 13h45 à 19h.

KOMELEC BAGNOLET « Professionnel »
M° GALLIENI - « Le Carnot » - 12, rue Sadi Carnot - 93170 Bagnolet
Tél. : (1) 43 63 64 64 - Fax : (1) 43 63 77 32

OUVERTURE PROCHAINE DE KOMELEC MARSEILLE

Commande mini. (sur papier libre) : 100 F - Administrations & Sociétés : bon de C[de] minimum : 1000 FHT. - Catalogue : 32 F (remboursés à la 1ère commande). Port : + 40 F jusqu'à 3 kg. 70 F en colissimo. Prix indicatifs. La présente liste de prix annule et remplace toutes les précédentes. Extrait de notre catalogue connectique.
LES PRIX S'ENTENDENT TTC

Quand on est le plus important revendeur d'Atari TT en France, on a d'autres arguments que celui de la baisse des prix…



1 **Service Après vente**
Grâce à notre clientèle exigeante, vous bénéficiez de réparations ultra rapides

2 **Vente par Correspondance**
Un service rapide, un règlement par carte bancaire, une expédition par chronopost

3 **Compétences**
Une concentration des connaissances sur les logiciels et matériels Atari

4 **Démonstration**
La plus importante salle de démonstration dédiée à de l'Atari Business Computer

5 **Stock**
La plus importante disponibilité de matériels et périphériques fonctionnant sur Atari

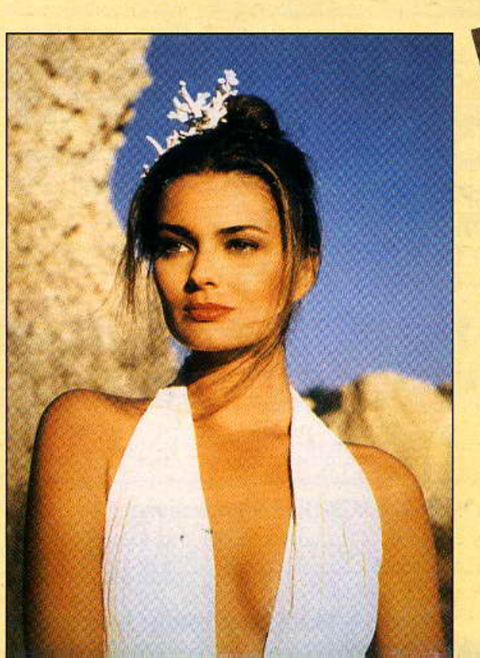
6 **Reprises,**
SCAP reprend aux meilleures conditions vos Atari ST pour tout achat de STE, Mega STE & TT.

7 **Ecrans Multi-synchro**
Ces écrans acceptent toutes les résolutions de vos Atari. Possibilité de reprise de vos anciens écrans.

SCAP Informatique

Pin's SCAP disponible

Document intégralement réalisé sur une station de pré-presse Atari TT avec les logiciels Calamus SL Retouche CD Didotlineart.



8 **Flashage**
4 photocomposeuses en service Vos documents Calamus en très haute résolution. Sortie couleur de vos mises en page Calamus.

9 **Réparations**
Notre service technique saura vous réparer, dans les meilleurs délais, vos ordinateurs.

10 **Domaine public**
Un catalogue complet de tous nos logiciels du domaine public. Recevez-le contre 25F timbres. 30F la disquette.

11 **Occasions**
SCAP propose le plus grand choix d'ordinateurs d'occasions. Ces machines sont révisées et garanties 6 mois à des prix défiant toute concurrence.

12 **NOUVEAUTÉS**
Grands écrans 1280x960 monochrome pour MEGA STE. Dynaccad 3D Devpac II Lattice C V.5 Il nous reste quelques STACY… N'hésitez pas à nous appeler pour plus d'informations.

SCAP Informatique

62, rue Gabriel Péri
93200 Saint-Denis
Tél: (1)42.43.22.78
Fax: (1)42.43.92.70

Publicité intégralement réalisée sur une station Atari TT, avec les logiciels Calamus SL, Didot Lineart, Retouche Professionnel, scanner Epson GT-6000 puis flashé sur photocomposeuse. Epreuve couleur réalisée sur AgfaProof.